CENDRe: Concept Extraction with Natural Domain Representations

Antonia Holzapfel* **Andres Felipe Posada Moreno** **Sebastian Trimpe**
Institute for Data Science in Mechanical Engineering (DSME)
RWTH Aachen University
Theaterstraße 35-39, 52062 Aachen, Germany

Abstract

Convolutional neural networks (CNNs) are widely used for time-series classification, but their deployment in critical contexts requires understanding the patterns across signal domains that drive their predictions (e.g., temporal or spectral). Concept extraction (CE) methods identify such patterns by analyzing representations within the models' latent space. However, existing time-series CE methods have three limitations: they operate only in the time domain and cannot locate features in other domains, they depend on a predefined number of concepts that biases them, and they produce localizations misaligned with the regions the model uses. We address these limitations with CENDRe, a concept extraction method for CNNs. For concept discovery, we propose a procedure that clusters per-timestep latent representations in two stages, with a silhouette-guided aggregation that selects the number of concepts automatically. For localization, we define a presence score that contrasts the latent representations with their prototypes, and use its gradients to produce masks that concentrate on the regions driving the concept. These gradients, propagated through a differentiable invertible mapping of the input such as a Fourier transform, yield localizations for the same concepts in different domains. Finally, each concept receives a relevance score that quantifies its contribution to each class. On synthetic benchmarks, CENDRe achieves representation correctness comparable to state-of-the-art CE methods and higher importance correctness, while producing visualizations in different domains. On real bearing-fault data, CENDRe extracts the frequency bands driving the model's predictions, located in regions commonly inspected for fault diagnosis, producing evidence to assess the model that time-domain CE methods cannot.

1 Introduction

Deep learning, particularly convolutional neural networks (CNNs), has become a widespread choice for time-series classification and regression. Applications range from condition monitoring and predictive maintenance on industrial equipment [22, 43] to early-warning and diagnostic systems in healthcare [23, 36, 11], settings where a wrong prediction can translate into costly misbehavior. In such domains, stakeholders cannot simply take a prediction at face value: they need explanations to verify that it is based on plausible signal features and not on spurious correlations. For such inspection, explanations must be both faithful to the model's behavior and expressed in an interpretable domain. Time-domain explanations cover at most half the story and other domains are often better suited to interpreting signals. Since models trained on time signals, especially CNNs, can learn features in the time, frequency, or other domains, explanation methods should also operate on these domains to make the corresponding features intelligible.

*Corresponding author: holzapfel@dsme.rwth-aachen.de

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Among XAI methods, concept methods [32] are post-hoc and grounded in the model's own latent space, capturing its global behavior. Specifically, Concept Extraction (CE) discovers recurring latent patterns that the model learned to distinguish across a dataset [21]. While original work in CE has focused on the image domain [12, 19, 33, 34], recent works have transferred CE to time series [16, 48]. However, these existing methods have three main limitations: First, their explanations live exclusively in the time domain, leaving unexplained the features that the model uses in any other relevant domain (e.g., frequency). Second, the number of concepts K must be fixed before extraction, forcing a manual sweep that biases the resulting explanations. Yet K reflects how the model decides and should be determined by the data, not the user. Third, their localization methods produce masks that may either miss regions the model uses [34, 16] or cover regions it does not [48, 47]. These limitations restrict the practical usefulness of existing CE time-series methods, as explanations in other domains are not covered, and strong priors on concept number and localization are included.

We address these three limitations with **CENDRe** (*Concept Extraction with Natural Domain Representations*), a CE method for time-series CNNs that extracts concepts in any natural domain, that is, any invertible and differentiable transform of the input in which signal content is directly interpretable, such as time or frequency. To this end, we propose a concept discovery algorithm that clusters timestep-wise activations in two stages, with silhouette-guided aggregation selecting the number of concepts automatically, rather than a user. Building on the discovered concepts, we introduce a localization mechanism based on the gradients of a contrastive presence score that compares the latent representations and concept prototypes, with respect to the input. These gradients produce masks that concentrate attribution on the regions influencing the concept. The same gradients, propagated through virtual inspection layers [44], yield concept localizations in other natural domains (for example, the frequency domain). We evaluate CENDRe on synthetic time-series with ground-truth concepts and on two natural bearing-fault datasets [25, 41]. Across all settings, CENDRe extracts concepts in both the time and frequency domains, while the baselines are inherently limited to the time domain. On time-domain concepts, CENDRe outperforms the baselines on importance correctness while recovering the same ground-truth concepts. On frequency-domain concepts, CENDRe recovers the ground-truth bands on synthetic data and extracts bands that include the characteristic fault frequencies experts use to identify each fault type on real bearing-fault data [25, 41].

To summarize, the main contributions in this work are:

1. **CENDRe.** We introduce a CE method for time-series CNNs that, to our knowledge, is the first to extract and visualize concepts in both the time and frequency domains.
2. **Concept localization mechanism that transfers across domains.** We backpropagate gradients of a novel contrastive presence score through differentiable invertible transforms of the input, yielding concept masks in any natural domain.
3. **Automatic concept number selection.** We develop a procedure that automatically estimates the number of relevant concepts K from the latent structure of the data, by overestimating and then aggregating clusters. This replaces the manual choice of K with the micro-cluster count $J \gg K$, to which the extraction is substantially less sensitive.
4. **Evaluation on synthetic and real datasets.** On synthetic time-series, CENDRe substantially outperforms prior CE baselines on importance correctness and uniquely recovers ground-truth concepts in the frequency domain alongside the time domain. On real bearing-fault datasets, it identifies the characteristic fault frequencies experts use for diagnosis.

2 Related Work

Concept-based explanations describe a model through human-interpretable patterns that correspond to directions or regions in its latent space [32]. Concept Extraction (CE) discovers such patterns automatically from data, without requiring a predefined example set. In the image domain, established CE methods include ACE [12], ECLAD [34], ConceptSHAP [47], and PACE [19]. In time-series, prior concept-based work has applied concept *testing* with TCAV [21] and manually curated concept sets [27, 10]. Only two recent works perform concept *extraction*: ECLAD-ts [16] clusters per-timestep local aggregated descriptors at a probe layer, and MultiVISION [48] clusters segments defined by the effective receptive field (ERF) of highly activated neurons. However, both methods are contrained to the time domain for explanations. CENDRe is a CE method for time-series CNNs that extracts concepts across natural domains, including the frequency domain.

Despite the central role of frequency domain analysis in signal processing and related areas [31, 3], no CE method operates in this domain. Spectral attribution methods, on the other hand, produce local explanations directly in the frequency domain either via perturbation or gradients. Perturbation methods in the frequency or time-frequency domain [8, 5] probe the model through targeted alterations of the spectrum, and virtual inspection layers [44] insert a differentiable, invertible transform into the forward graph so that attribution can be propagated to any chosen signal representation. These methods are local, explaining individual predictions rather than global model behavior. CENDRe reformulates the virtual inspection layer mechanism to bring global concept explanations into the frequency domain, which, to our knowledge, has not been done before.

In CE the number of concepts K is typically fixed manually through sweeps or visual inspection [12, 19, 48, 34, 16]. In general clustering, by contrast, silhouette analysis has long been proposed as a criterion for selecting K [37], alongside alternatives such as density-based methods [6], which we include as variants. To our knowledge, CENDRe is the first CE method to determine the number of concepts automatically, guided by silhouette scores, and thus reducing the inductive bias introduced by manual concept number selection.

To localize concepts in the input, existing time-series CE methods typically rely on two heuristics: either upsampling low-resolution activation maps to the input size [34, 16], or taking the full ERF of the probe layer as a concept patch [48]. Upsampling attributes importance to the aggregation position even though its activation is driven by the full ERF, while the full-ERF patch can include regions the model did not meaningfully rely on. CENDRe avoids both caveats by introducing visualizations based on gradients with respect to the latent representation of the concept, propagated through virtual inspection layers, so that the attribution is concentrated on the regions that influence the concept, yielding compact localizations across domains.

3 Concept Extraction with Natural Domain Representations (CENDRe)

CENDRe is a CE method for time-series CNNs that produces explanations across natural domains with a data-driven number of concepts. Realizing this rests on three technical contributions: (i) discovering the number of concepts K from the data rather than imposing it, (ii) localizing each concept on the regions the model actually uses, and (iii) extending these localizations to natural domains beyond the input. The paragraphs below recall the per-timestep latent representation we build on, address each of these challenges, and conclude with a class-wise concept-importance score.

Latent representation via LADs. For concept discovery, we adapt LADs [34, 16], timestep-wise aggregations of activations across layers, as they capture patterns at multiple levels of abstraction and are approximately position-invariant thanks to the translation equivariance of CNNs. For building such a per-timestep descriptor, we first consider a trained 1D CNN $f : \mathbb{R}^{T\times D} \to \mathbb{R}^{n_{\text{out}}}$ for supervised classification or regression on multivariate time-series of length T with D channels and n_{out} targets. We select a subset $\mathcal{L}$ of its internal layers as probe layers. Concept discovery operates on a dataset $\mathcal{D} \subset \mathbb{R}^{T\times D}$. For each input sample $x \in \mathcal{D}$, $a_\ell(x) \in \mathbb{R}^{T_\ell \times D_\ell}$ denotes the activation map at probe layer $\ell \in \mathcal{L}$. We linearly upsample each activation map to the longest probe length T^* via f_U. For each $x \in \mathcal{D}$ and timestep $t \in \{0, \dots, T^* - 1\}$, the LAD $z_t(x)$ is the per-timestep slice of the concatenated upsampled maps,

$$z_t(x) \;=\; \big[\, f_U(a_{\ell_1}(x))[t] \;\dots\; f_U(a_{\ell_{|\mathcal{L}|}}(x))[t] \,\big] \in \mathbb{R}^{D^*}, \qquad D^* \coloneqq \textstyle\sum_{\ell\in\mathcal{L}} D_\ell. \tag{1}$$

Thus, similar LADs correspond to similar local evidence across the selected layers, and the collection of LADs over the dataset is the input to the first clustering stage.

Concept discovery via clustering. CENDRe discovers latent concepts by clustering LADs $z_t(x)$ with agglomerative or density-based methods, which select the number of concepts K automatically and can capture non-spherical clusters expected in high-dimensional latent spaces. Applying such methods directly to all LADs is infeasible at scale, so we first compress LADs to micro-centroids, and then cluster those micro-centroids into concepts.

In the first stage, mini-batch k-means on LADs produces a set of $J \gg K$ micro-centroids $\Gamma = \{\mu_j\}_{j=0}^{J-1}$ with $\mu_j \in \mathbb{R}^{D^*}$, and we assign each LAD to its nearest micro-centroid,

$$\pi(x,t) \;\coloneqq\; \underset{j\in\{0,\dots,J-1\}}{\arg\min} \; \|z_t(x) - \mu_j\|_2^2. \tag{2}$$

In the second stage, we cluster the J micro-centroids into K concepts via a mapping $\kappa : \{0, \dots, J-1\} \to \{0, \dots, K-1\}$, so each concept $C_k = \{\mu_j \mid \kappa(j) = k\}$ is a group of micro-centroids capturing similarly encoded patterns.

By default, we use silhouette-guided hierarchical aggregation under cosine distance, with K the value maximizing the average silhouette score across micro-centroids (CENDRe$_{\text{silhouette}}$) [29, 37]. We additionally evaluate two variants: k-means [39] directly on LADs with predefined K (CENDRe$_{k\text{Means}}$), and HDBSCAN [6] on the micro-centroids (CENDRe$_{\text{HDBSCAN}}$). See App. C.3 for hyperparameters.

Concept masks. We localize each concept on each input through gradients with respect to an interpretable domain. Gradients concentrate the attribution on regions that relate to the concept, rather than covering the full receptive field or shrinking past the relevant evidence within it. Pulling the same gradient through any invertible differentiable transform allows the visualization of the concept in the corresponding domain (e.g. the input domain or the frequency domain).

We propose a presence score, defined below, which captures what makes a LAD belong to concept C_k by contrasting its similarity to the assigned micro-centroid with the mean similarity to micro-centroids of the other concepts. To express this contrast, we partition the micro-centroid indices into the in-concept set $\mathcal{J}_k \coloneqq \kappa^{-1}(k) \subseteq \{0, \dots, J-1\}$ and its complement $\bar{\mathcal{J}}_k \coloneqq \{0, \dots, J-1\} \setminus \mathcal{J}_k$, and we collect the timesteps where the LAD of x is in C_k in $T_k(x) \coloneqq \{t \in \{0, \dots, T^*-1\} \mid \pi(x,t) \in \mathcal{J}_k\}$. To assess how similar a LAD is to a micro-centroid, we use the inverse-decay similarity $d(z, \mu) \coloneqq (1 + \|z - \mu\|_2)^{-1} \in (0, 1]$ between a LAD z and a micro-centroid μ. For every $t \in T_k(x)$, the presence score is then

$$\rho_k(x,t) \;\coloneqq\; d\big(z_t(x),\, \mu_{\pi(x,t)}\big) \;-\; \frac{1}{|\bar{\mathcal{J}}_k|} \sum_{j \in \bar{\mathcal{J}}_k} d\big(z_t(x),\, \mu_j\big). \tag{3}$$

A high presence score indicates strong concept membership: the LAD is close to its assigned micro-centroid and far from those of the other concepts. We aggregate it over $T_k(x)$ to obtain a concept presence per sample $\bar{\rho}_k(x) \coloneqq |T_k(x)|^{-1} \sum_{t \in T_k(x)} \rho_k(x,t)$, with $\bar{\rho}_k(x) = 0$ when $T_k(x) = \emptyset$.

To obtain a mask in an arbitrary domain we follow *virtual inspection layers* [44]: a fixed, differentiable, invertible transform ϕ is inserted into the computational graph as $\phi^{-1} \circ \phi$ for attribution only, leaving the model's forward mapping unchanged. We instantiate ϕ in two ways: the frequency domain ($\phi = \mathcal{F}$), where $\mathcal{F} : \mathbb{R}^{T \times D} \to \mathbb{C}^{F \times D}$ is the real-input fast Fourier transform (rFFT) with $F \coloneqq \lfloor T/2 \rfloor + 1$ and $x_f \coloneqq \mathcal{F}(x)$, and the time domain, where ϕ is the identity. App. D.1.2 instantiates two further choices in time-frequency domains. Through the inserted inverse transform, $\bar{\rho}_k$ becomes a function of x_f, and $\nabla_{x_f} \bar{\rho}_k$ denotes its gradient at $x_f = \phi(x)$. The masks become the sensitivities of $\bar{\rho}_k$ to the input and to the log-power-spectrum, respectively (derivation in App. A),

$$m_k^{\text{time}}(x) \coloneqq \text{ReLU}\big(\nabla_x \bar{\rho}_k(x)\big), \qquad m_k^{\text{freq}}(x) \coloneqq \text{ReLU}\big(\Re\big(\overline{x_f} \odot \nabla_{x_f} \bar{\rho}_k(x)\big)\big). \tag{4}$$

The ReLU keeps the components at which amplifying the signal, or its spectral power, increases the concept's presence. Negative components mark evidence against the concept and are discarded for localization. Although LADs are indexed on the upsampled grid $t \in \{0, \dots, T^*-1\}$, masks are obtained by backpropagation to the original input $x \in \mathbb{R}^{T \times D}$ or its rFFT image $x_f \in \mathbb{C}^{F \times D}$. For visualization, a noise tunnel is applied on the masks [40] and they are smoothed by a moving average along the domain axis.

Concept importance scores. For each class c, concept C_k, and input channel ch, we ask whether the presence of C_k on channel ch pushes the model toward or away from predicting c, and by how much. The importance score $I_{c,k,\text{ch}} \in [-1, 1]$ encodes this: magnitude reports the strength of the effect. The sign on its own carries no meaning, but opposite signs across classes indicate that the concept pushes those classes apart. We compute $I_{c,k,\text{ch}}$ from the gradient of the per-class score y_c with respect to the input, weighted by the concept mask of C_k on channel ch. For classification, $y_c(x)$ is the logit $f(x)_c$. For regression, $y_c(x)$ is the c-th regression output.

We measure the instance-level relevance as the overlap between the regions where the concept lives and the regions to which y_c is sensitive. Let $m_{k,\text{ch}}^{\text{dom}}(x)$ and $s_{c,\text{ch}}^{\text{dom}}(x)$ denote the channel slices of the mask and the sensitivity of y_c in domain $\text{dom} \in \{\text{time}, \text{freq}\}$. We write $x_{\text{ch}} \in \mathbb{R}^T$ and $x_{f,\text{ch}} \in \mathbb{C}^F$ for the time- and frequency-domain channels of x. The relevance is then their inner product over the corresponding domain axis, $r_{c,k,\text{ch}}(x) \;\coloneqq\; \big\langle m_{k,\text{ch}}^{\text{dom}}(x),\; s_{c,\text{ch}}^{\text{dom}}(x) \big\rangle$.

The sensitivity $s^{\mathrm{dom}}_{c,\mathrm{ch}}(x)$ is obtained by backpropagation in the chosen domain: $s^{\mathrm{time}}_{c,\mathrm{ch}}(x) = \partial y_c(x)/\partial x_{\mathrm{ch}}$, and $s^{\mathrm{freq}}_{c,\mathrm{ch}}(x) = \Re\big(\overline{x_{f,\mathrm{ch}}} \odot \nabla_{x_{f,\mathrm{ch}}} y_c(x)\big)$ (App. A). We smooth $s^{\mathrm{dom}}_{c,\mathrm{ch}}(x)$ with a noise tunnel to damp single-point gradient noise. Then, we average $r_{c,k,\mathrm{ch}}(x)$ over $\mathcal{D}_{k,\mathrm{ch}} \coloneqq \{\, x \in \mathcal{D} \mid m^{\mathrm{dom}}_{k,\mathrm{ch}}(x) \neq 0 \,\}$, the samples on which C_k leaves a non-zero mask on channel ch, to obtain a dataset-level estimate of the concept's effect. If $\mathcal{D}_{k,\mathrm{ch}} = \emptyset$, we set the expectation to 0. We divide the result by the largest such average across all (c', k', ch') to make entries comparable on $[-1, 1]$:

$$I_{c,k,\mathrm{ch}} \;=\; \frac{\mathbb{E}_{x\sim\mathcal{D}_{k,\mathrm{ch}}}[\, r_{c,k,\mathrm{ch}}(x)\,]}{\max_{c',k',\mathrm{ch}'}\big|\mathbb{E}_{x\sim\mathcal{D}_{k',\mathrm{ch}'}}[\, r_{c',k',\mathrm{ch}'}(x)\,]\big| + \varepsilon},$$

for a small $\varepsilon > 0$.

To obtain a single, class-agnostic importance value per concept–channel pair (as required by downstream metrics), we contrast the most-impacted class against the mean of the rest,

$$I_{k,\mathrm{ch}} \;=\; 0.5\left|I_{c^\star(k,\mathrm{ch}),\,k,\mathrm{ch}} \;-\; \frac{1}{n_{\mathrm{out}}-1}\sum_{c\neq c^\star(k,\mathrm{ch})} I_{c,k,\mathrm{ch}}\right| \;\in\; [0,1], \tag{5}$$

with $c^\star(k,\mathrm{ch}) \coloneqq \arg\max_c |I_{c,k,\mathrm{ch}}|$. For $n_{\mathrm{out}} = 1$, we set $I_{k,\mathrm{ch}} \coloneqq |I_{0,k,\mathrm{ch}}|$. Values close to 1 identify concepts that selectively drive the prediction for at least one class, whereas values close to 0 mark concepts whose contribution is either negligible or uniform across classes.

4 Experimental design

A CE method should identify the recurring patterns a model learns, how they relate to the input, and how they influence the predictions. Yet, there is no direct ground truth on what a CNN relies on. Thus, we evaluate CE methods on synthetic datasets where each input is annotated with the masks of its class-discriminative primitives, assuming a model trained on these datasets relies on them. A faithful CE method should produce concepts tied to those primitives (*representation*) and assign higher importance to concepts on discriminative primitives than to unrelated ones (*importance*).

We compare CE methods by running each through multiple trained models and measuring representation and importance on the resulting concepts. In an experimental run, we train a CNN on a synthetic dataset, run the CE method on the same data to obtain concepts with time- and frequency-domain masks and per-class importance scores, and score these concepts on the validation data against the primitive masks. We repeat across datasets and seeds, then compare CE methods on representation and importance correctness on the synthetic data, and qualitatively on the real bearing-fault datasets CWRU and BearingPD by inspecting how concepts relate to characteristic fault frequencies.

4.1 Evaluation metrics

To evaluate a CE method, we assess how much the resulting concepts align with the primitives through two correctness scores: *representation*, measuring how well concept masks align with primitive masks, and *importance*, measuring whether each concept's importance score reflects its primitive's function in the dataset. To this end, we propose two modifications over the evaluation procedure of [34, 16]: We score concept alignment with the *Relevance Mass Accuracy* rather than the EDT score, and we compute the correctness metrics using *soft* variants (sRC, sIC) that avoid the *sensitivity to an alignment threshold* of the original Representation and Importance Correctness scores.

Alignment. For the alignment comparison between concepts and primitives to make sense, all masks must be on the same scale. CENDRe masks $m^{\mathrm{dom}}_k(x)$ can take any real value, so we normalize them per instance to $\tilde{m}^{\mathrm{dom}}_k(x) \in [0,1]$ by clamping negatives to zero and dividing by the maximum, while baseline masks are already bounded or binary. On the ground-truth side, $g^{\mathrm{dom}}_p(x) \in \{0,1\}$ is the binary mask of primitive p, with n_p primitives and K extracted concepts in total. We score the alignment with the *Relevance Mass Accuracy* (RMA) [2], which for a continuous mask $u \in \mathbb{R}^{|\mathrm{dom}|\times D}$ and binary reference $v \in \{0,1\}^{|\mathrm{dom}|\times D}$ (with $|\mathrm{time}| = T$ and $|\mathrm{freq}| = F$) measures the fraction of u's positive energy that falls inside v,

$$\mathrm{RMA}(u,v) = \frac{\sum_{t,\mathrm{ch}} \mathrm{ReLU}(u_{t,\mathrm{ch}}) \cdot \mathbf{1}[v_{t,\mathrm{ch}} > 0]}{\sum_{t,\mathrm{ch}} \mathrm{ReLU}(u_{t,\mathrm{ch}}) + \varepsilon} \;\in\; [0,1], \tag{6}$$

where $\varepsilon > 0$ penalizes empty masks. We choose RMA over the EDT score because it accepts continuous masks directly and tolerates models that use only part of a primitive's support.

Representation and Importance metrics. With the concept and primitive alignments computed per instance, we aggregate them into one score per property. Following [16], we treat each pair (k, ch) of concept and channel as its own evaluation unit, and define its alignment to a pair (p, ch) of primitive and channel as the average RMA over the dataset, $A_{p,k,\mathrm{ch}} \coloneqq \mathbb{E}_{x\sim\mathcal{D}}[\mathrm{RMA}(\tilde{m}^{\mathrm{dom}}_{k,\mathrm{ch}}(x),\, g^{\mathrm{dom}}_{p,\mathrm{ch}}(x))] \in [0, 1]$. For each primitive, *Soft Representation Correctness* (sRC) picks the concept whose alignment summed across channels is highest, and averages this over primitives and channels. *Soft Importance Correctness* (sIC) instead weights each concept's class-agnostic importance $I_{k,\mathrm{ch}}$ from Eq. (5) by its best alignment to any primitive,

$$\mathrm{sRC} \coloneqq \frac{1}{n_p D} \sum_{p} \max_{k} \sum_{\mathrm{ch}} A_{p,k,\mathrm{ch}}, \qquad \mathrm{sIC} \coloneqq \frac{\sum_{k,\mathrm{ch}} \big(\max_p A_{p,k,\mathrm{ch}}\big)\, I_{k,\mathrm{ch}}}{\sum_{k,\mathrm{ch}} |I_{k,\mathrm{ch}}|} \;\in\; [0, 1]. \tag{7}$$

4.2 Experimental setup

Models and training. We train three 1D CNN architectures: InceptionTime10 [18], ResNet1D-18 [13], and DenseNet1D-121 [17]. Each model takes the standard hyperparameters of its reference implementation, with full configurations listed in App. C.1. We optimize the negative log-likelihood using AdamW [26] (learning rate 10^{-4}, weight decay 10^{-7}) with early stopping on the validation loss (patience 20, up to 500 epochs). Each dataset is split 80/20 with balanced sampling, and every training configuration is repeated over 11 seeds (0–10).

Datasets. We use two synthetic binary-classification families with known ground-truth primitive masks: SYNTHETICLOCAL, which injects localized temporal shapes (square, circle, triangle) into a noisy sinusoidal baseline, and SYNTHETICFREQUENCY, which places class-discriminative peaks in specific frequency bands (mid, mid-high, high). For real-world signals we use *CWRU Bearing* [41], a rolling-element bearing fault dataset with four classes (healthy plus three fault locations) and two accelerometer channels (fan end and drive end). Both channels are sampled at $f_s = 12\,\mathrm{kHz}$ and segmented into windows of $T = 2048$ samples with stride 1024 (50% overlap). The full dataset list is in Table 1 of App. B, with additional synthetic variants and a second bearing-fault benchmark, which are evaluated in App. D.

Concept extraction hyperparameters. All CE methods are implemented in PyTorch [1] and require probe layers per model, listed in Table 4 of App. C. Each method runs on 256 samples with the same random seed as the corresponding model training, is fit on the model's training partition and evaluated on its validation partition. The methods differ in how the number of concepts K is set. For fixed-K methods (ECLAD-ts, MultiVISION, CENDRe$_{k\mathrm{Means}}$) we sweep over $K \in \{2, 4, 6, 8, 10\}$ and report the configuration with the highest sRC. For methods that determine K automatically (CENDRe$_{\mathrm{silhouette}}$, CENDRe$_{\mathrm{HDBSCAN}}$) we take 50 micro-clusters, a choice whose effect the sensitivity analysis in App. E quantifies. CENDRe's noise tunnel uses 5 samples with Gaussian perturbations of standard deviation 0.01 for instance masks, 0.1 for logit sensitivities, and a moving average of 5 for visualizations. [2]

5 Empirical results

We report results on the SYNTHETICLOCAL and SYNTHETICFREQUENCY dataset families, and on the natural bearing-fault dataset CWRU. The experimental setup follows Section 4, with further evaluations in App. D. The main findings of our experiments are: (i) On SYNTHETICLOCAL, CENDRe extracts concept regions that match the ground truth qualitatively, with representation correctness similar to ECLAD-ts but higher importance correctness. (ii) Frequency-domain concepts on SYNTHETICFREQUENCY localize in the ground-truth bands, with high representation and importance correctness. (iii) Concepts extracted from CWRU separate cleanly by class and align with characteristic fault frequencies, evidencing that CENDRe transfers from controlled synthetic settings to real signals.

[2] The code is freely available at https://github.com/antoHolz/CENDRe/

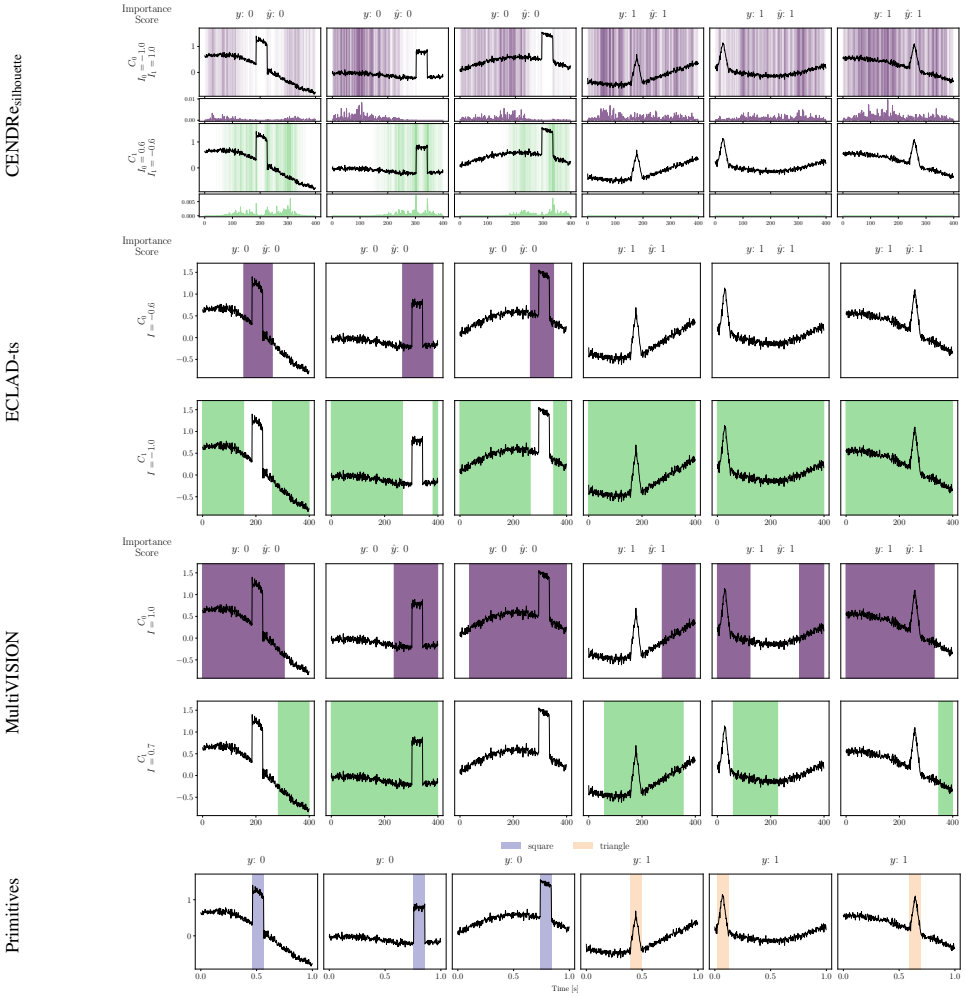


Figure 1: Concepts extracted from a ResNet1D-18 trained on a SYNTHETICLOCAL square-triangle dataset. Within each grid, rows are concepts, columns are samples, and row labels show importance scores. The bottom row shows the ground-truth primitive locations for the same samples. CENDRe and ECLAD-ts extract the same input features, while MultiVISION shows no coherent pattern.

Synthetic datasets. The SYNTHETICLOCAL family tests whether the compared CE methods extract the local cues time-series CNNs rely on for classification. Figure 1 shows CE on a representative square-triangle variant from this family. ECLAD-ts highlights the complete square defined by the ground-truth primitive. CENDRe picks up the same regions with importance scores of similar magnitude, but weights the corners more strongly, where the signal differs most recognizably from the baseline. MultiVISION, by contrast, returns no coherent pattern. Overall, CENDRe extracts the local cues without tuning the number of concepts, and additionally highlights the most important structures within each concept. Figure 7 of App. D.1.1 repeats this comparison on two further architectures and the two remaining CENDRe variants, with the same outcome.

The quantitative metrics in Figure 2 (top row) are consistent with this qualitative picture. For the CENDRe$_{\text{silhouette/HDBSCAN}}$ variants, sRC is slightly lower than ECLAD-ts's on SYNTHETICLOCAL (k-means is essentially on par). Two factors plausibly contribute. First, the model exploits only a portion of the ground-truth primitive (a shortcut-learning pattern), and CENDRe$_{\text{silhouette/HDBSCAN}}$ stays faithful to that partial use, while ECLAD-ts's broader masks cover the full primitive. Second, k-means and ECLAD-ts are reported at the K with best sRC, which biases them toward human-recognizable concepts, whereas silhouette and HDBSCAN choose K from the latent structure itself. The latter is also visible in sIC, which is higher for the CENDRe$_{\text{silhouette/HDBSCAN}}$ variants, meaning their masks better match the model's per-timestep importance signal. MultiVISION underperforms

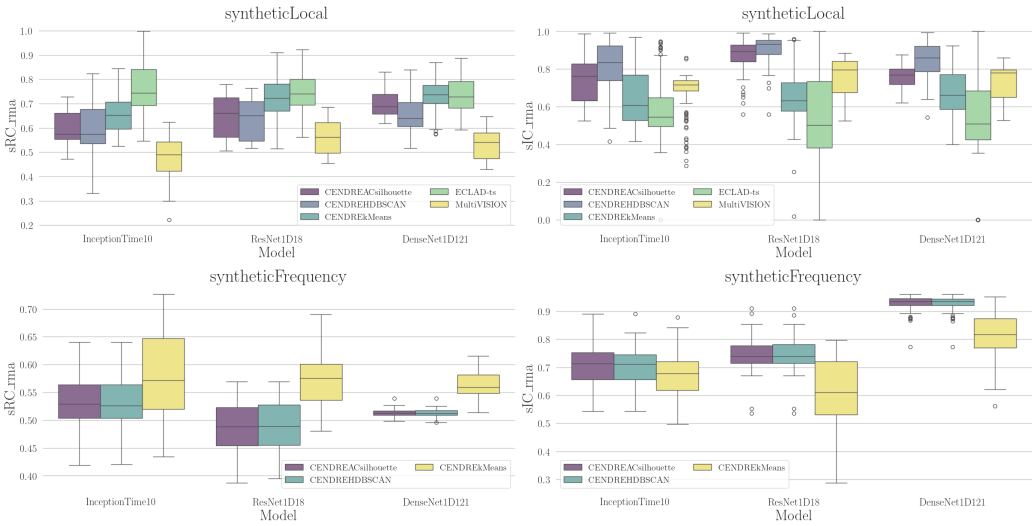


Figure 2: Boxplots of sRC (left) and sIC (right) on SYNTHETICLOCAL (top) and SYNTHETICFREQUENCY (bottom); higher is better. On SYNTHETICLOCAL, CENDRe achieves sRC close to ECLAD-ts and higher sIC, while MultiVISION is weak on both metrics. On SYNTHETICFREQUENCY, CENDRe variants reach sRC and sIC values close to those on SYNTHETICLOCAL.

on both metrics, in line with the lack of coherent concept structure observed qualitatively. The same ranking holds on the multichannel and confounded variants of the family (Figure 10 of App. D.1.5). Per-dataset values and pairwise significance tests are in App. D.1.3 and App. D.1.4. Across CENDRe variants the differences are small qualitatively (the three variants point to the same regions). Quantitatively CENDRe$_{\text{silhouette}}$ and CENDRe$_{\text{HDBSCAN}}$ perform similarly, while CENDRe$_{k\text{Means}}$ trades slightly higher sRC for slightly lower sIC, consistent with the K-selection effect noted above. We favor the silhouette variant because it reaches comparable quality in a single run without a K sweep, and picks a usable K across datasets out of the box. It is also the more robust of the two to its remaining hyperparameter, the number of micro-clusters J. Over three orders of magnitude, J explains 0.02 of the sRC variance under CENDRe$_{\text{silhouette}}$ and 0.14 under CENDRe$_{\text{HDBSCAN}}$, against 0.70 for the random seed (App. E), supporting our claim that silhouette-guided aggregation removes the dependence on a user-set number of concepts.

The SYNTHETICFREQUENCY family tests whether CE methods extract spectral cues exploited by the model. Among the compared methods, only CENDRe extends to the frequency domain, through its virtual inspection layers, so the baselines are omitted from this analysis. Figure 3 shows that the time-domain masks have no interpretable pattern, whereas the frequency-domain masks concentrate concept mass at specific bands, both per-sample and in the global concept correspondence. In the `midhighFreq-highFreq` variant shown, C_0 aligns with a single high-frequency band, while C_1 mixes mid-high and near-zero content, reflecting how the model composes information from several frequency components. When a model relies on frequency-related features, CENDRe captures them at both instance and global scale, turning uninterpretable time-domain masks into interpretable frequency explanations. Quantitatively, Figure 2 (bottom row) shows the same CENDRe behavior as the local case. CENDRe$_{\text{silhouette/HDBSCAN}}$ variants reach slightly lower sRC than k-means but higher sIC. Mean sRC is slightly below SYNTHETICLOCAL's, while sIC is comparable. CENDRe thus extracts concepts in the frequency domain at quality close to the time domain, showing that the localization transfers across domains.

Natural datasets. On CWRU (Figure 4), concepts separate mostly by class. C_0 aligns with a sharp, dominant peak at $\sim$2000 Hz on the drive-end channel, and shows weaker activity at $\sim$250 and $\sim$1000 Hz on the fan-end channel. Concepts C_2 and C_3 correspond more weakly to bands around 800–1000 Hz on the drive-end channel and on the fan-end channel, respectively. These specific bands give domain experts the information needed to assess whether the model is making predictions for the right frequency reasons. Analogous results on BearingPD (App. D.2.2) and on five datasets of the

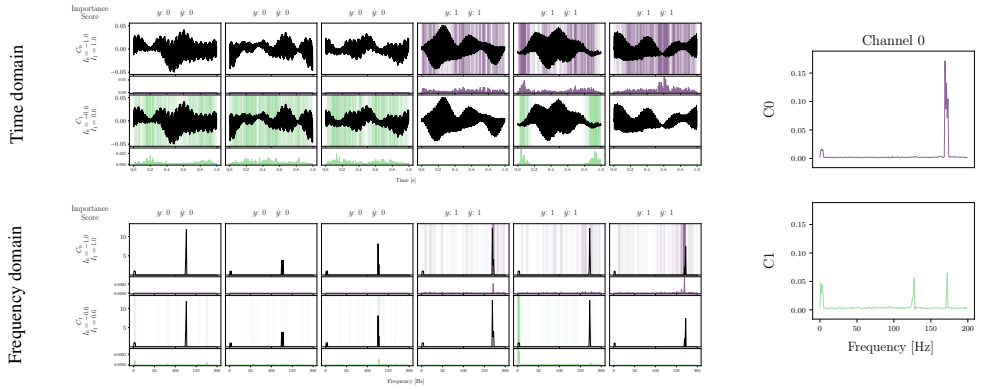


Figure 3: Concepts extracted from a ResNet1D-18 trained on a SYNTHETICFREQUENCY `midhighFreq-highFreq` dataset. The two left grids show time-domain (top) and frequency-domain (bottom) masks; within each, rows are concepts, columns are samples, and row labels show importance scores. The right panel shows the global concept-frequency correspondence, obtained by aggregating the frequency masks across samples. Time-domain masks show no interpretable pattern, whereas frequency-domain masks localize in specific bands, as confirmed by the correspondence plot.

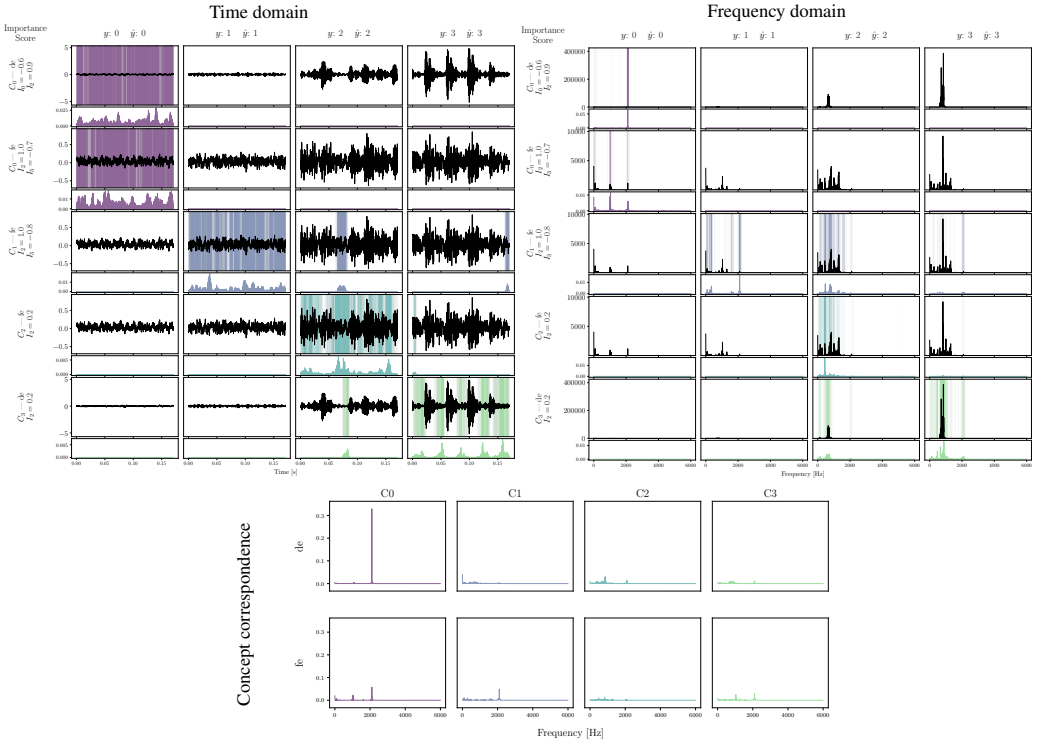


Figure 4: Concepts extracted from an InceptionTime10 trained on CWRU. The top grids show time-domain (left) and frequency-domain (right) masks; rows are concepts on channels with high importance and columns are samples. The bottom panel shows the concept-frequency correspondence per channel, with a fan end (fe) and drive end (de) channel. Each concept localizes almost exclusively on one class, with the frequency masks exposing characteristic fault frequencies.

UCR archive [9] (App. D.2.1) support that CENDRe transfers from controlled synthetic settings to real signals across domains.

6 Conclusion

We have presented CENDRe, a concept extraction method for time-series CNNs that addresses the three limitations of prior time-series CE methods. First, by propagating gradients of our contrastive presence score through virtual inspection layers, CENDRe extracts concepts in both time and frequency domains. The frequency masks recover the ground-truth bands on synthetic data and locate concept-specific bands on the natural bearing-fault datasets, where they support verification by domain experts. Second, our silhouette-guided aggregation removes the need to fix the number of concepts by hand and delivers results comparable to the best configuration of a manual K sweep in a single run. This replaces the hyperparameter K with the much less sensitive micro-cluster count J. Third, our gradient-based localization concentrates on the regions that drive the concept, yielding representation correctness comparable to prior methods and higher importance correctness on the synthetic benchmarks.

Being the first CE method to combine time and frequency domain concepts, CENDRe opens new avenues for future research. The same pipeline can be applied to other invertible representations such as time-frequency distributions. Architectures, by contrast, remain a current limitation: CENDRe's per-timestep LADs rely on the approximate translation equivariance of 1D CNNs. Extending the latent representation to architectures without this property, such as transformers or state-space models, is a natural next step. Furthermore, the discovered concepts could feed back into the development cycle, for instance to guide model debugging or to flag suspicious reliance on shortcut cues. Domain-specific user studies, assessing whether CENDRe's masks help experts within their own field, are also a natural direction for future work.

Acknowledgments and Disclosure of Funding

We thank Franziska Hoffmann for her valuable help in designing the synthetic datasets.

In addition, model training and experiments were partially performed with computing resources granted by RWTH Aachen University, under projects p0021919 and p0027430.

This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 520460745.

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and from the German Federal Ministry of Research, Technology and Space (BMFTR), the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW) and the Hessian Ministry of Science and Research, Arts and Culture (HMWK) under grant agreement No 101250682.

References

[1] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '24, page 929–947. Association for Computing Machinery, 2024. doi: 10.1145/3620665.3640366.

[2] Leila Arras, Ahmed Osman, and Wojciech Samek. Clevr-xai: A benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40, 2022.

[3] Karl Johan Åström and Richard Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2021.

[4] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[5] Thea Brüsch, Kristoffer K Wickstrøm, Mikkel N Schmidt, Tommy S Alstrøm, and Robert Jenssen. Explaining time series models using frequency masking. *arXiv preprint arXiv:2406.13584*, 2024. doi: 10.48550/arXiv.2406.13584.

[6] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[7] Francesca Campolongo, Jessica Cariboni, and Andrea Saltelli. An effective screening design for sensitivity analysis of large models. *Environmental modelling & software*, 22(10):1509–1518, 2007.

[8] Hyunseung Chung, Sumin Jo, Yeonsu Kwon, and Edward Choi. Time is not enough: Time-frequency based explanation for time-series black-box models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 394–403, 2024. doi: 10.1145/3627673.3679844.

[9] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Hexagon-ML Batista, Gustavo. The ucr time series classification archive, October 2018.

[10] Thomas Decker, Michael Lebacher, and Volker Tresp. Explaining Deep Neural Networks for Bearing Fault Detection with Vibration Concepts. In *INDIN*, pages 1–6, 2023. doi: 10.1109/INDIN51400.2023.10218170.

[11] Riccardo Fratti, Niccolò Marini, Manfredo Atzori, Henning Müller, Cesare Tiengo, and Franco Bassetto. A multi-scale cnn for transfer learning in semg-based hand gesture recognition for prosthetic devices. *Sensors*, 24(22):7147, 2024.

[12] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards Automatic Concept-based Explanations. In *NeurIPS*, volume 32. Curran Associates, Inc., 2019.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[14] Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. Classification of time series by shapelet transformation. *Data mining and knowledge discovery*, 2014.

[15] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[16] Antonia Holzapfel, Andres Felipe Posada Moreno, and Sebastian Trimpe. Concept extraction for time series with eclad-ts. In *World Conference on Explainable Artificial Intelligence*, pages 90–112. Springer, 2025.

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

[18] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

[19] Vidhya Kamakshi, Uday Gupta, and Narayanan C Krishnan. PACE: Posthoc Architecture-Agnostic Concept Extractor for Explaining CNNs. In *IJCNN*, pages 1–8, 2021. doi: 10.1109/IJCNN52387.2021.9534369.

[20] Dave S Kerby. The simple difference formula: An approach to teaching nonparametric correlation. *Comprehensive Psychology*, 3:11–IT, 2014.

[21] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *ICML*, pages 2668–2677. PMLR, 2018.

[22] Yejin Kim and Young-Keun Kim. Time-frequency multi-domain 1d convolutional neural network with channel-spatial attention for noise-robust bearing fault diagnosis. *Sensors*, 23(23): 9311, 2023.

[23] Dimitri Kraft and Peter Rumm. Atrial fibrillation and atrial flutter detection using deep learning. *Sensors*, 25(13):4109, 2025.

[24] Gregory Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O'Leary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019.

[25] Christian Lessmeier, James Kuria Kimotho, Detmar Zimmer, and Walter Sextro. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In *PHM society European conference*, volume 3, 2016.

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[27] Anders Gjølbye Madsen, William Theodor Lehn-Schiøler, Áshildur Jónsdóttir, Bergdís Arnardóttir, and Lars Kai Hansen. Concept-based explainability for an eeg transformer model. In *MLSP*, pages 1–6. IEEE, 2023.

[28] Max D Morris. Factorial sampling plans for preliminary computational experiments. *Quality control and applied statistics*, 37(6):307–310, 1992.

[29] Daniel Müllner. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011.

[30] Stephen Olejnik and James Algina. Generalized eta and omega squared statistics: measures of effect size for some common research designs. *Psychological methods*, 8(4):434, 2003.

[31] Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

[32] Eleonora Poeta, Gabriele Ciravegna, Eliana Pastor, Tania Cerquitelli, and Elena Baralis. Concept-based explainable artificial intelligence: A survey. *ACM Computing Surveys*, 2023.

[33] Andrés Felipe Posada-Moreno, Lukas Kreisköther, Tassilo Glander, and Sebastian Trimpe. Scale-preserving automatic concept extraction (SPACE). *Machine Learning*, 112(11):4495–4525, 2023. doi: 10.1007/s10994-023-06373-2.

[34] Andrés Felipe Posada-Moreno, Nikita Surya, and Sebastian Trimpe. ECLAD: Extracting Concepts with Local Aggregated Descriptors. *Pattern Recognition*, 147:110146, 2023.

[35] Robert B Randall and Jerome Antoni. Rolling element bearing diagnostics—a tutorial. *Mechanical systems and signal processing*, 25(2):485–520, 2011.

[36] Swathy Ravi and Ashalatha Radhakrishnan. A hybrid 1d cnn-bilstm model for epileptic seizure detection using multichannel eeg feature fusion. *Biomedical physics & engineering express*, 10 (3):035040, 2024.

[37] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[38] Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[39] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010.

[40] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[41] Wade A. Smith and Robert B. Randall. Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. *Mechanical Systems and Signal Processing*, 64–65:100–131, 2015.

[42] Ilya M Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280, 2001.

[43] Emmanuel Stathatos, Evangelos Tzimas, Panorios Benardos, and George-Christopher Vosniakos. Convolutional neural networks for raw signal classification in cnc turning process monitoring. *Sensors*, 24(5):1390, 2024.

[44] Johanna Vielhaben, Sebastian Lapuschkin, Grégoire Montavon, and Wojciech Samek. Explainable ai for time series via virtual inspection layers. *Pattern Recognition*, 150:110309, 2024.

[45] Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics: Methodology and distribution*, pages 196–202. Springer, 1992.

[46] Moritz Wolter, Felix Blanke, Jochen Garcke, and Charles Tapley Hoyt. Ptwt-the pytorch wavelet toolbox. *Journal of Machine Learning Research*, 25(80):1–7, 2024.

[47] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. In *NeurIPS*, volume 33, pages 20554–20565, 2020.

[48] Raneen Younis, Sergej Zerr, and Zahra Ahmadi. Multivariate time series analysis: An interpretable cnn-based model. In *International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE, 2022.

A Derivation of the spectral concept mask

This appendix derives the frequency-domain CENDRe formulas of Sec. 3 as the sensitivity of a real scalar functional to the log-power spectrum. The same derivation applies both to the spectral concept mask (setting $E = \bar{\rho}_k$) and to the frequency-domain logit sensitivity (setting $E = y_c(\cdot)$).

Setup. Let $x_f = \mathcal{F}(x) \in \mathbb{C}^{F\times D}$ be the rFFT of the time-domain signal x, and let $E : \mathbb{C}^{F\times D} \to \mathbb{R}$ be any real-valued, differentiable scalar functional. Writing $x_f = u + iv$ with $u = \Re(x_f)$ and $v = \Im(x_f)$, the power spectrum is $P = |x_f|^2 = u^2 + v^2 \in \mathbb{R}^{F\times D}_{\geq 0}$. All operations below are element-wise over the $F \times D$ grid.

Wirtinger preliminaries. Treating x_f and $\overline{x_f}$ as formally independent, the Wirtinger derivatives are

$$\partial_{x_f} E \;\coloneqq\; \tfrac{1}{2}\big(\partial_u - i\,\partial_v\big)E, \qquad \partial_{\overline{x_f}} E \;\coloneqq\; \tfrac{1}{2}\big(\partial_u + i\,\partial_v\big)E.$$

For real-valued E we have $\partial_{x_f} E = \overline{\partial_{\overline{x_f}} E}$. Following the convention adopted by standard automatic-differentiation libraries (e.g. PyTorch), we set $\nabla_{x_f} E \coloneqq \partial_{\overline{x_f}} E$, so that the direction of real steepest ascent in the (u, v) parametrization is $2\nabla_{x_f} E$.

Phase-fixed power perturbation. Writing $x_f = re^{i\theta}$ with $r = \sqrt{P}$ and holding the phase θ fixed,

$$\frac{dx_f}{dP} \;=\; \frac{e^{i\theta}}{2\sqrt{P}} \;=\; \frac{x_f}{2P}, \qquad \frac{d\overline{x_f}}{dP} \;=\; \frac{\overline{x_f}}{2P}.$$

Chain rule. The complex chain rule gives

$$\frac{dE}{dP} \;=\; \partial_{x_f} E \odot \frac{dx_f}{dP} + \partial_{\overline{x_f}} E \odot \frac{d\overline{x_f}}{dP} \;=\; \frac{1}{2P}\Big(\overline{\partial_{\overline{x_f}} E} \odot x_f + \partial_{\overline{x_f}} E \odot \overline{x_f}\Big),$$

where the second equality uses $\partial_{x_f} E = \overline{\partial_{\overline{x_f}} E}$. Since the two terms in the bracket are complex conjugates of each other,

$$\frac{dE}{dP} \;=\; \frac{1}{P}\,\Re\big(\overline{x_f} \odot \nabla_{x_f} E\big).$$

Log-power form. Multiplying through by P yields the sensitivity with respect to $\log P$, which avoids the $1/P$ singularity at empty bins and gives a dimensionless quantity:

$$\frac{dE}{d\log P} \;=\; \Re\big(\overline{x_f} \odot \nabla_{x_f} E\big). \tag{8}$$

CENDRe spectral mask and logit sensitivity. Substituting $E = \bar{\rho}_k(x_f)$ into (8) and keeping only the positive contributions recovers the spectral concept mask of Sec. 3,

$$m_k^{\text{freq}}(x) \;=\; \mathrm{ReLU}\big(\Re(\overline{x_f} \odot \nabla_{x_f} \bar{\rho}_k(x_f))\big).$$

Similarly, setting $E = y_c(x)$ yields the frequency-domain logit sensitivity $s_{c,\text{ch}}^{\text{freq}}(x) = \Re(\overline{x_{f,\text{ch}}} \odot \nabla_{x_{f,\text{ch}}} y_c(x))$ used in the relevance score.

B Synthetic Dataset Details

We construct four synthetic binary-classification time-series datasets with known ground-truth concept masks (primitives), summarized in Table 1. SYNTHETICLOCAL and SYNTHETICFREQUENCY are reported in the main paper, while SYNTHETICLMC and SYNTHETICLCONF are included only here, as additional tests for per-channel attribution and confounder robustness.

Table 1: The four synthetic datasets. Each is instantiated as several variants pairing primitives or sweeping a confounder ratio.

Name	D	Variants	Role
SYNTHETICLOCAL	1	9	Localized shapes
SYNTHETICFREQUENCY	1	6	Frequency bands
SYNTHETICLMC	2	6	Per-channel attribution
SYNTHETICLCONF	1	6	Confounder robustness

Each variant contains 1,000 samples of length $T = 400$. A base signal is generated by superimposing low-frequency sinusoidal components with additive Gaussian noise, then class-discriminative patterns are injected at random, non-overlapping positions. Binary ground-truth masks $g_p^{\text{dom}} \in \{0,1\}^{|\text{dom}| \times D}$ (with $|\text{time}| = T$ for time-domain patterns and $|\text{freq}| = F = \lfloor T/2 \rfloor + 1$ for frequency-domain patterns) record the exact locations of each injected pattern per channel, plus the complementary "absence" masks described below. Shared generation parameters are summarized in Table 2.

Negatives as primitives. A model can learn "absence of square" or "absence of the frequency band" as a concept just as readily as their presence. We therefore treat the absence of every involved primitive as a primitive of its own, with a ground-truth mask covering the regions where the primitive is absent. This ensures that the correctness scores reward concepts that capture absence as well as presence.

B.1 SYNTHETICLOCAL — Localized Shape Patterns

SYNTHETICLOCAL uses three primitive shapes of length 40:

- **Square**: a constant-amplitude block.
- **Triangle**: a triangular envelope that peaks at the center and decays linearly to zero at both ends.

Table 2: Shared generation parameters for all synthetic datasets.

Parameter	Value
Sample length T	400
Samples per dataset	1 000
Class balance	50/50
Overlap allowed	No
*Shape datasets (*SYNTHETICLOCAL / SYNTHETICLMC / SYNTHETICLCONF*)*	
Base signal	Noisy sinusoidal (f_s=1000 Hz)
Base noise level	0.05
Base components	2 sinusoids, $[0.5,\ 2]$ Hz
Shape length	40 time steps
Shape amplitude	1.0
*Frequency datasets (*SYNTHETICFREQUENCY*)*	
Base signal	Frequency-domain generator (f_s=400 Hz)
Base noise level	0.02
Base components	8 sinusoids, $[1,\ 5]$ Hz
Peaks per band	5
Peak amplitude	$\mathcal{U}[1.5,\ 2.0]$
Random phase	Yes

- **Circle**: a semi-elliptical (half-circle) envelope, producing a smooth bump.

Each shape is injected at a uniformly random position, avoiding overlap with previously placed patterns. Nine variants (D=1) span three regimes:

- *Paired* (3 variants). Each class is characterized by a different shape. **Purpose:** discriminate two class-specific shapes in a clean setting. Variants: `square-circle`, `square-triangle`, `triangle-circle`.
- *Single-shape vs nothing* (3 variants). One class contains a single shape, the other contains only the noisy base signal. **Purpose:** probe presence detection of a single primitive against a background. Variants: `square-nothing`, `triangle-nothing`, `circle-nothing`.
- *Disjunctive vs nothing* (3 variants). One class contains either of two shapes (chosen at random per sample), the other contains only the noisy base signal. **Purpose:** test whether the method recovers a concept defined as a disjunction of primitives. Variants: `squareorcircle-nothing`, `squareortriangle-nothing`, `triangleorcircle-nothing`.

Figure 5 shows an example from the `square-circle` variant.

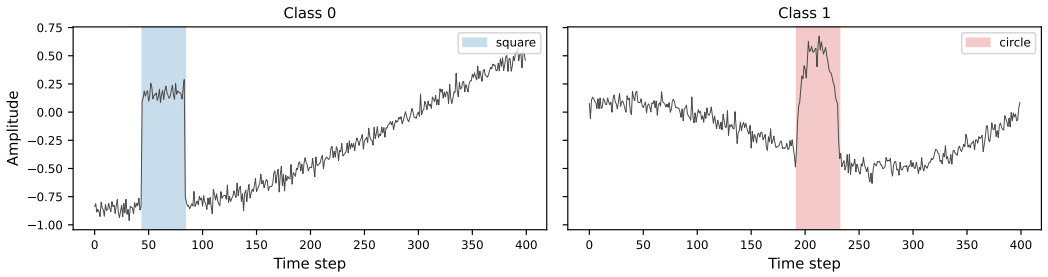


Figure 5: Example samples from the SYNTHETICLOCAL `square-circle` variant. Class 0 contains a square pattern (blue region) and class 1 a circle pattern (red region), each injected at a random position into a noisy sinusoidal base signal.

B.2 SYNTHETICFREQUENCY — Frequency Band Patterns

SYNTHETICFREQUENCY uses frequency-domain patterns drawn from three bands:

- **midFreq**: $[70,\ 75]$ Hz
- **midhighFreq**: $[125,\ 130]$ Hz
- **highFreq**: $[170,\ 175]$ Hz

Each band is realized by injecting 5 sinusoidal peaks with amplitudes drawn from $\mathcal{U}[1.5,\ 2.0]$ and random phases into the discrete Fourier transform of the base signal, then inverting back to the time domain. Ground-truth masks are defined in the frequency domain, marking the bins within each band. Six variants (D=1) span two regimes:

- *Pure paired* (3 variants). Each class is characterized by a different band, with no other peaks. **Purpose:** clean spectral discrimination, the frequency analogue of the paired regime in SYNTHETICLOCAL. Variants: `midFreq-highFreq`, `midFreq-midhighFreq`, `midhighFreq-highFreq`.
- *Paired + confounder* (3 variants). Each pure-paired base is augmented with the third (unused) band injected at a $50/50$ ratio, equally likely in both classes. **Purpose:** test whether the method ignores a spectrally prominent but uninformative band and focuses on the causal pair. Variants: `midFreq-highFreq_conf-midhighFreq-50-50`, `midFreq-midhighFreq_conf-highFreq-50-50`, `midhighFreq-highFreq_conf-midFreq-50-50`.

Figure 6 shows an example from the `midFreq-highFreq` variant.

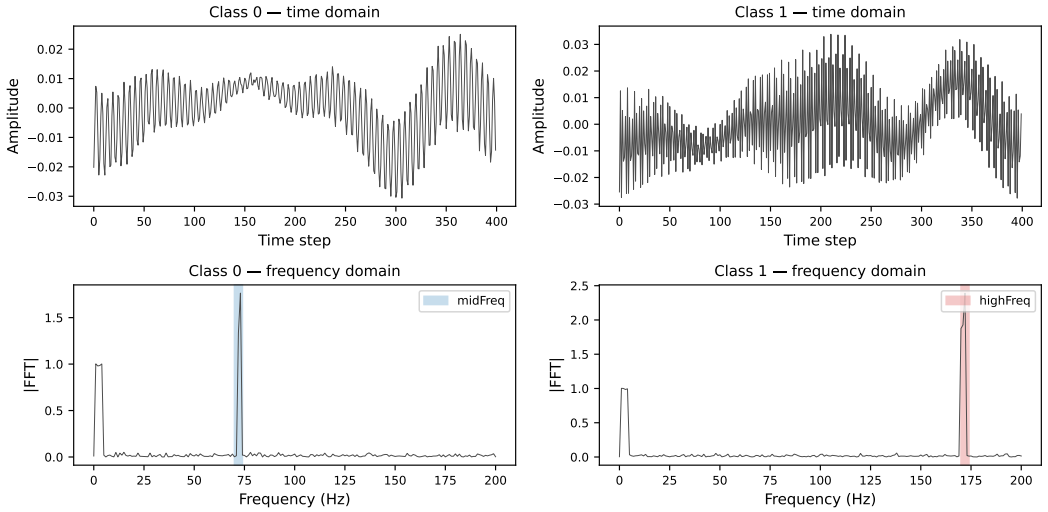


Figure 6: Example samples from the SYNTHETICFREQUENCY `midFreq-highFreq` variant. Top: time-domain signals. Bottom: FFT magnitudes with ground-truth frequency bands highlighted (midFreq in blue, highFreq in red). Class 0 contains the midFreq pattern, class 1 the highFreq pattern.

We deliberately keep SYNTHETICFREQUENCY single-channel. With discriminative frequencies present in some channels and absent in others, the cross-channel structure is dominated by the low-frequency base signal, and a CNN trained on such a setup is too likely to learn local time-domain shapes induced by these residuals rather than the discriminative bands themselves. The synthetic benchmarks are only informative when model behavior is strongly predictable from the dataset construction, a property that is already strained on simpler datasets, which the multi-channel frequency setting would defeat.

B.3 SYNTHETICLMC — Multi-Channel concept localization

SYNTHETICLMC reuses the three shape primitives of SYNTHETICLOCAL but signals are two-channel (D=2). Six variants in the following format exist:

- *Random-channel single-shape* (3 variants). One class contains a single shape injected into a uniformly chosen channel, the other contains only the noisy base signal in both

channels. **Purpose:** test whether the method recovers a channel-agnostic concept, defined by shape rather than channel index. Variants: `square-nothing_ch2_random`, `triangle-nothing_ch2_random`, `circle-nothing_ch2_random`.

- *Specific-channel paired* (3 variants). Each of two class-discriminative shapes is assigned to a designated channel, one shape per channel. **Purpose:** test per-channel attribution, whether the method assigns each concept to the correct channel rather than mixing evidence across them. Variants: `square-circle_ch2_specificChannels`, `square-triangle_ch2_specificChannels`, `triangle-circle_ch2_specificChannels`.

B.4 SYNTHETICLCONF — Confounder Robustness

SYNTHETICLCONF reuses the three shape primitives of SYNTHETICLOCAL (D=1) and adds a non-discriminative confounder shape on top of each paired base, with controlled class–confounder correlation. Six variants combine two paired bases with three confounder ratios:

- *Bases.* `square-circle` with `triangle` as confounder, and `triangle-circle` with `square` as confounder.
- *Ratios.* $40/60$, $50/50$, and $60/40$, where the ratio gives the proportion of confounder presence in class 0 versus class 1.

Purpose: test robustness to spurious correlations. At $50/50$ the confounder carries no label information, while at $40/60$ and $60/40$ it does, so a method that latches onto the prominent shape rather than the causal pair is penalized more strongly at the off-balance ratios.

C Implementation details

This appendix collects the implementation details needed to reproduce CENDRe and the baselines: backbone architectures, the probe layers used for concept extraction, and the CENDRe clustering hyperparameters.

C.1 Model architectures

Table 3 lists the exact hyperparameters of the three 1D CNN architectures used throughout the experiments (§4.2). All three are direct 1D adaptations of the corresponding standard image-classification architectures, with 2D convolutions replaced by 1D convolutions and 2D pooling by 1D pooling.

Table 3: Architecture hyperparameters used for the three 1D CNN backbones. Defaults follow each architecture's reference implementation.

Model	Configuration
InceptionTime10 [18]	10 inception modules, 32 filters per branch, base kernel size 40, with bottleneck layer and residual connections every three modules.
ResNet1D-18 [13]	Four stages with $[2, 2, 2, 2]$ residual blocks, expansion factor 1.
DenseNet1D-121 [17]	Block configuration $[6, 12, 24, 16]$, growth rate 32, compression factor 0.5, 64 initial features.

C.2 Probe layers

The choice of which layers to analyze constitutes the most critical hyperparameter for CE. Table 4 reports the specific layers selected for each network. In ECLAD-ts, probe layers are sampled at uniform intervals throughout the architecture. MultiVISION instead relies on a single bottleneck layer, which prevents the analysis of blocks that residual connections might bypass. CENDRe adopts the same probe layers as ECLAD-ts. For MultiVISION, the neuron activation extraction threshold is set to the 0.99 quantile of the layer under examination, and activations are grouped via k-means clustering.

Table 4: Layers used for CE with each method and model.

Model	ECLAD-ts and CENDRe
InceptionTime10	`model.inception_block.inception_layers.`n_b`.bottleneck`, $n_b \in \{6, 7, 8, 9\}$
ResNet1D-18	`model.layers.`n_b`.1.relu` $n_b \in \{0, 1, 2, 3\}$
DenseNet1D-121	`model.features.transition`n_b`.conv`, $n_b \in \{1, 2, 3\}$; `model.features.denseblock4.block.15.conv2`

Model	MultiVISION
InceptionTime10	`model.inception_block.inception_layers.9.bottleneck`
ResNet1D-18	`model.layers.2.1.relu`
DenseNet1D-121	`model.features.transition3.conv`

C.3 CENDRe clustering hyperparameters

The second stage of $\text{CENDRe}_{\text{silhouette}}$ and $\text{CENDRe}_{\text{HDBSCAN}}$ runs on the $J = 50$ micro-centroids returned by mini-batch k-means. $\text{CENDRe}_{\text{silhouette}}$ uses average linkage and selects $K \in \{2, \ldots, 10\}$ by the silhouette score, both evaluated under cosine distance. $\text{CENDRe}_{\text{HDBSCAN}}$ uses $\text{min_cluster_size} = 5$ under cosine distance. Micro-centroids that HDBSCAN labels as noise are excluded from $\bar{\mathcal{J}}_k$ when computing ρ_k.

D Additional results

This appendix extends Section 5 in two directions. §D.1 broadens the synthetic evaluation: it compares all three CENDRe variants across three architectures, visualizes concepts under two further invertible transforms, breaks the quantitative comparison down per dataset with significance tests, and extends it to two dataset families that test multichannel localization and confounder robustness. §D.2 extends the qualitative evaluation to five real-world datasets from the UCR archive and to a second bearing-fault dataset, BearingPD, on two architectures.

D.1 Synthetic datasets

We extend the synthetic evaluation of Section 5 along four lines. We first repeat the qualitative comparison of Figure 1 across all three CENDRe variants and three architectures. We then visualize concepts under two further invertible transforms, the short-time Fourier and the wavelet transform (§D.1.2). Next, we break the family-level quantitative comparison of Figure 2 down per dataset and test the resulting ordering for statistical significance (§D.1.3, §D.1.4). Finally, we extend the quantitative comparison to two appendix-only dataset families, SYNTHETICLMC and SYNTHETICLCONF, testing multichannel localization and confounder robustness.

D.1.1 Qualitative comparison across CENDRe variants and architectures

Section 5 reports the silhouette-guided aggregation variant ($\text{CENDRe}_{\text{silhouette}}$) as the default, extracted from a single ResNet1D-18. Figure 7 extends that view in two directions, adding the two remaining CENDRe variants and repeating the extraction on two further architectures. It shows the concepts extracted by the five CE methods (rows) from a ResNet1D-18, an InceptionTime10, and a DenseNet1D-121 (columns) trained on the same SYNTHETICLOCAL square-triangle dataset as Figure 1. $\text{CENDRe}_{\text{silhouette}}$ and $\text{CENDRe}_{\text{HDBSCAN}}$ set the number of concepts automatically; for $\text{CENDRe}_{k\text{Means}}$, ECLAD-ts, and MultiVISION we report the run whose number of concepts maximizes the RC score. For space, each panel shows the two most important concepts.

The three CENDRe variants stay close to each other within every column, pointing to similar regions with comparable importance scores. Reporting $\text{CENDRe}_{\text{silhouette}}$ alone in the main text therefore loses little. Against the baselines, the comparison reproduces the one in Section 5 on all three architectures. ECLAD-ts marks the square as a uniform block and its complement as a second concept, while CENDRe spreads graded mass over similar regions, with the largest values at the edges of the shapes. MultiVISION splits the signal into broad contiguous segments whose boundaries follow neither primitive.

All three architectures yield concepts that map onto the primitives, with slight differences between them. On the ResNet1D-18 and the InceptionTime10, every method separates the discriminative shapes from the background. The DenseNet1D-121 follows the same pattern, with the CENDRe variants splitting the background further into additional low-importance concepts, left out of the figure for space.

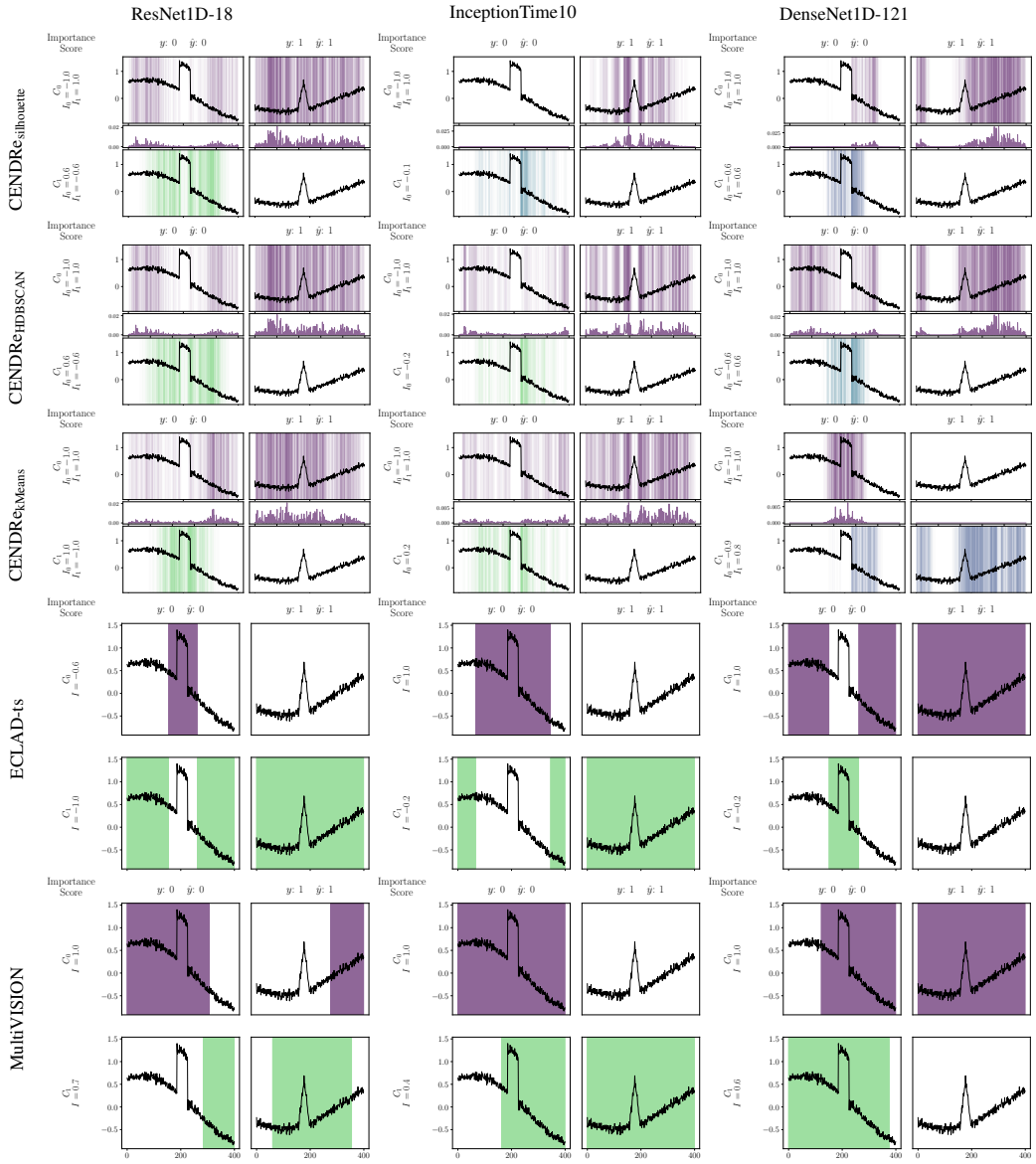


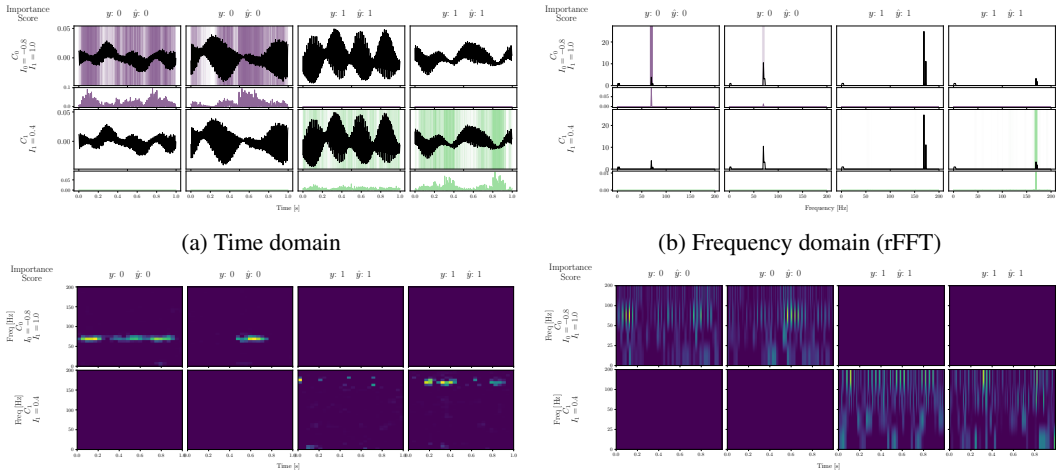
Figure 7: Concepts extracted by CENDRe$_{\text{silhouette}}$, CENDRe$_{\text{HDBSCAN}}$, CENDRe$_{\text{kMeans}}$, ECLAD-ts, and MultiVISION (rows) from a ResNet1D-18, an InceptionTime10, and a DenseNet1D-121 (columns, in that order) trained on the same SYNTHETICLOCAL square-triangle dataset as Figure 1. For space, each panel shows the two most important concepts; within a panel, rows are concepts, columns are one class-0 and one class-1 sample, and row labels report the importance scores. Across architectures, ECLAD-ts isolates the square as a solid block, CENDRe recovers the same regions as ECLAD-ts but as graded masks whose largest values partially sit at the edges, and MultiVISION shows no correspondence to either primitive. On DenseNet1D-121 the CENDRe variants extract additional low-importance concepts (not shown) that further partition the background.

D.1.2 Visualizations in other natural domains

In Section 3, we instantiate the transform ϕ as a Fourier transform to visualize the concepts in the frequency domain, but mention how other invertible transforms could be used instead. While we don't test them quantitatively, such transforms are straightforward to implement and can be used to visualize the same concepts in other domains, e.g., wavelet transforms or short-time Fourier transforms for time-frequency analysis. We have implemented such transforms by creating virtual inspection layers for short-time Fourier transform (STFT) and discrete wavelet transform (DWT).

The STFT builds on the implementation of PyTorch [1], and the DWT on the differentiable filter banks of the PyTorch Wavelet Toolbox [46], which in turn builds on PyWavelets [24]. The STFT $\mathcal{S} : \mathbb{R}^{T\times D} \to \mathbb{C}^{F\times N\times D}$ applies the rFFT to overlapping, windowed segments of length W with hop size H, producing $F \coloneqq \lfloor W/2 \rfloor + 1$ frequency bins across N time frames. Because $\mathcal{S}$ is complex-valued like $\mathcal{F}$, the sensitivity in Eq. 4 carries over unchanged after substituting $x_s \coloneqq \mathcal{S}(x)$ for x_f. Its derivation only requires a complex transform output, not a specific transform (App. A). We invert $\mathcal{S}$ exactly by choosing an analysis window that satisfies the nonzero-overlap-add condition. The DWT $\mathcal{W} : \mathbb{R}^{T\times D} \to \mathbb{R}^{T\times D}$ decomposes x into approximation and detail coefficients through an orthogonal wavelet filter bank. This filter bank admits an exact inverse by construction. Since $\mathcal{W}$ is real-valued, unlike $\mathcal{F}$ and $\mathcal{S}$, its mask reduces to the plain ReLU gradient, following the same pattern as m_k^{time} in Eq. 4. We selected STFT and DWT because both are exactly invertible, matching the requirement virtual inspection layers place on ϕ. Other time-frequency transforms, such as the continuous wavelet transform, lack an exact inverse and are left for future work.

In Figure 8 we show qualitative results of reports in the time-frequency domain using such transforms. Across the four views, both concepts land where the ground truth places them, but the choice of ϕ decides how granular that visualization is. The FFT already isolates C_0 in the midFreq band, while its C_1 mask spreads over a wide range with no dominant peak. The STFT resolves C_1 into the highFreq band and shows it active at isolated time frames rather than across the whole record. Comparing C_0 in the FFT and STFT views shows that the lower activation magnitude of the concept corresponds with it being active in a shorter timespan. The DWT recovers the same time structure, but its highest band spans 100 to 200 Hz, so the highFreq band cannot be separated within it.



(a) Time domain

(b) Frequency domain (rFFT)

(c) Time-frequency domain (STFT)

(d) Time-frequency domain (wavelet)

Figure 8: The same two concepts, extracted with CENDRe$_{\text{silhouette}}$ from an InceptionTime10 trained on the SYNTHETICFREQUENCY `midFreq-highFreq` variant, visualized under four choices of the transform ϕ. The time-domain masks (a) carry no interpretable pattern, while all three spectral views place C_0 in the ground-truth midFreq band at $[70, 75]$ Hz. The rFFT (b) resolves that band as a sharp peak and also places C_1 in the highFreq band at $[170, 175]$ Hz, though only on one of the class 1 samples. The STFT (c) highlights C_1 in both class one samples, and shows that it is active only at isolated time frames, whereas C_0 persists across the full record. The DWT (d) reproduces the same time structure at coarser frequency resolution, since its highest dyadic band spans 100 to 200 Hz and cannot separate the highFreq band inside it.

D.1.3 Quantitative evaluation: summary per dataset

In Figure 2 of Section 5, each box summarizes the sRC and sIC values across all datasets of a given family. Tables 5 and 6 report the same runs resolved per dataset, as mean $\pm$ standard deviation with a 95% confidence interval over the 33 runs per cell (3 architectures $\times$ 11 seeds), with the best method per dataset in bold. The interval is the standard 95% Student-t confidence interval for the mean,

$\bar{y} \pm t_{0.975,\,n-1}\, s/\sqrt{n}$. The family-level ranking of Figure 2 holds dataset by dataset and is not an artifact of pooling.

Table 5: Distributional summary (mean $\pm$ std, 95% CI) of sRC across CE methods and datasets.

	CENDRe$_{\text{silhouette}}$		CENDRe$_{\text{HDBSCAN}}$		CENDRe$_{k\text{Means}}$		ECLAD-ts		MultiVISION	
	mean $\pm$ std	95% CI	mean $\pm$ std	95% CI	mean $\pm$ std	95% CI	mean $\pm$ std	95% CI	mean $\pm$ std	95% CI
SYNTHETICLOCAL										
circle-nothing	0.731 $\pm$ 0.035	[0.718, 0.743]	0.716 $\pm$ 0.030	[0.705, 0.726]	0.774 $\pm$ 0.049	[0.756, 0.791]	**0.824 $\pm$ 0.060**	**[0.803, 0.846]**	0.584 $\pm$ 0.070	[0.559, 0.609]
square-circle	0.599 $\pm$ 0.053	[0.581, 0.618]	0.561 $\pm$ 0.035	[0.549, 0.574]	0.662 $\pm$ 0.066	[0.639, 0.686]	**0.723 $\pm$ 0.094**	**[0.690, 0.756]**	0.450 $\pm$ 0.052	[0.431, 0.468]
triangle-circle	0.622 $\pm$ 0.073	[0.596, 0.648]	0.575 $\pm$ 0.061	[0.553, 0.597]	0.672 $\pm$ 0.087	[0.642, 0.703]	**0.689 $\pm$ 0.099**	**[0.654, 0.724]**	0.443 $\pm$ 0.057	[0.423, 0.463]
square-nothing	0.710 $\pm$ 0.028	[0.701, 0.720]	0.711 $\pm$ 0.034	[0.699, 0.724]	0.757 $\pm$ 0.046	[0.741, 0.774]	**0.813 $\pm$ 0.057**	**[0.792, 0.833]**	0.600 $\pm$ 0.045	[0.584, 0.616]
square-triangle	0.597 $\pm$ 0.066	[0.573, 0.620]	0.563 $\pm$ 0.049	[0.545, 0.580]	0.639 $\pm$ 0.062	[0.617, 0.661]	**0.674 $\pm$ 0.087**	**[0.643, 0.704]**	0.453 $\pm$ 0.039	[0.439, 0.467]
triangle-nothing	0.742 $\pm$ 0.054	[0.723, 0.762]	0.729 $\pm$ 0.053	[0.710, 0.747]	0.804 $\pm$ 0.059	[0.783, 0.825]	**0.838 $\pm$ 0.061**	**[0.817, 0.860]**	0.609 $\pm$ 0.031	[0.598, 0.620]
squareorcircle-nothing	0.617 $\pm$ 0.054	[0.598, 0.637]	0.620 $\pm$ 0.035	[0.607, 0.633]	0.669 $\pm$ 0.064	[0.646, 0.691]	**0.705 $\pm$ 0.046**	**[0.688, 0.721]**	0.526 $\pm$ 0.042	[0.511, 0.541]
triangleorcircle-nothing	0.623 $\pm$ 0.063	[0.601, 0.646]	0.618 $\pm$ 0.069	[0.592, 0.643]	0.677 $\pm$ 0.067	[0.653, 0.700]	**0.701 $\pm$ 0.040**	**[0.687, 0.715]**	0.525 $\pm$ 0.040	[0.511, 0.540]
squareortriangle-nothing	0.618 $\pm$ 0.060	[0.597, 0.640]	0.600 $\pm$ 0.073	[0.573, 0.627]	0.684 $\pm$ 0.073	[0.658, 0.710]	**0.703 $\pm$ 0.040**	**[0.689, 0.717]**	0.533 $\pm$ 0.035	[0.521, 0.546]
Mean	0.651 $\pm$ 0.078	[0.642, 0.660]	0.633 $\pm$ 0.082	[0.623, 0.642]	0.704 $\pm$ 0.084	[0.695, 0.714]	**0.741 $\pm$ 0.091**	**[0.731, 0.751]**	0.525 $\pm$ 0.077	[0.516, 0.534]
SYNTHETICFREQUENCY										
midFreq-highFreq	0.503 $\pm$ 0.042	[0.488, 0.518]	0.504 $\pm$ 0.042	[0.489, 0.519]	**0.593 $\pm$ 0.058**	**[0.573, 0.614]**	–	–	–	–
midFreq-highFreq _conf-midhighFreq-50-50	0.490 $\pm$ 0.050	[0.472, 0.508]	0.490 $\pm$ 0.049	[0.473, 0.508]	**0.554 $\pm$ 0.054**	**[0.535, 0.573]**	–	–	–	–
midFreq-midhighFreq	0.532 $\pm$ 0.034	[0.520, 0.544]	0.532 $\pm$ 0.034	[0.520, 0.544]	**0.585 $\pm$ 0.052**	**[0.567, 0.603]**	–	–	–	–
midFreq-midhighFreq _conf-highFreq-50-50	0.503 $\pm$ 0.036	[0.490, 0.516]	0.502 $\pm$ 0.038	[0.489, 0.516]	**0.570 $\pm$ 0.047**	**[0.553, 0.586]**	–	–	–	–
midhighFreq-highFreq	0.546 $\pm$ 0.037	[0.533, 0.559]	0.546 $\pm$ 0.036	[0.533, 0.559]	**0.583 $\pm$ 0.047**	**[0.566, 0.599]**	–	–	–	–
midhighFreq-highFreq _conf-midFreq-50-50	0.488 $\pm$ 0.031	[0.477, 0.499]	0.489 $\pm$ 0.030	[0.478, 0.500]	**0.540 $\pm$ 0.049**	**[0.523, 0.558]**	–	–	–	–
Mean	0.510 $\pm$ 0.044	[0.504, 0.517]	0.511 $\pm$ 0.044	[0.504, 0.517]	**0.571 $\pm$ 0.054**	**[0.563, 0.578]**	–	–	–	–

Table 6: Distributional summary (mean $\pm$ std, 95% CI) of sIC across CE methods and datasets.

	CENDRe$_{\text{silhouette}}$		CENDRe$_{\text{HDBSCAN}}$		CENDRe$_{k\text{Means}}$		ECLAD-ts		MultiVISION	
	mean $\pm$ std	95% CI	mean $\pm$ std	95% CI	mean $\pm$ std	95% CI	mean $\pm$ std	95% CI	mean $\pm$ std	95% CI
SYNTHETICLOCAL										
circle-nothing	0.806 $\pm$ 0.091	[0.774, 0.839]	**0.850 $\pm$ 0.100**	**[0.814, 0.886]**	0.666 $\pm$ 0.140	[0.616, 0.716]	0.684 $\pm$ 0.283	[0.584, 0.784]	0.686 $\pm$ 0.137	[0.637, 0.734]
square-circle	0.721 $\pm$ 0.127	[0.676, 0.766]	**0.816 $\pm$ 0.140**	**[0.767, 0.866]**	0.605 $\pm$ 0.098	[0.570, 0.639]	0.550 $\pm$ 0.250	[0.461, 0.638]	0.735 $\pm$ 0.120	[0.692, 0.777]
triangle-circle	0.794 $\pm$ 0.123	[0.751, 0.838]	**0.849 $\pm$ 0.121**	**[0.806, 0.893]**	0.653 $\pm$ 0.145	[0.602, 0.705]	0.474 $\pm$ 0.298	[0.369, 0.580]	0.743 $\pm$ 0.135	[0.695, 0.791]
square-nothing	0.799 $\pm$ 0.103	[0.763, 0.836]	**0.873 $\pm$ 0.090**	**[0.840, 0.905]**	0.668 $\pm$ 0.135	[0.620, 0.716]	0.466 $\pm$ 0.234	[0.383, 0.549]	0.711 $\pm$ 0.078	[0.683, 0.739]
square-triangle	0.766 $\pm$ 0.097	[0.731, 0.800]	**0.842 $\pm$ 0.111**	**[0.803, 0.881]**	0.626 $\pm$ 0.110	[0.587, 0.665]	0.596 $\pm$ 0.228	[0.515, 0.677]	0.731 $\pm$ 0.104	[0.694, 0.768]
triangle-nothing	0.784 $\pm$ 0.125	[0.740, 0.828]	**0.873 $\pm$ 0.115**	**[0.832, 0.914]**	0.661 $\pm$ 0.174	[0.599, 0.723]	0.599 $\pm$ 0.287	[0.497, 0.701]	0.741 $\pm$ 0.083	[0.712, 0.771]
squareorcircle-nothing	0.819 $\pm$ 0.098	[0.784, 0.854]	**0.870 $\pm$ 0.094**	**[0.835, 0.905]**	0.712 $\pm$ 0.128	[0.667, 0.758]	0.551 $\pm$ 0.193	[0.483, 0.619]	0.707 $\pm$ 0.100	[0.672, 0.743]
triangleorcircle-nothing	0.812 $\pm$ 0.098	[0.778, 0.847]	**0.884 $\pm$ 0.084**	**[0.853, 0.915]**	0.676 $\pm$ 0.123	[0.633, 0.720]	0.558 $\pm$ 0.305	[0.450, 0.666]	0.739 $\pm$ 0.065	[0.717, 0.762]
squareortriangle-nothing	0.827 $\pm$ 0.097	[0.792, 0.861]	**0.869 $\pm$ 0.095**	**[0.834, 0.904]**	0.651 $\pm$ 0.128	[0.606, 0.697]	0.523 $\pm$ 0.217	[0.446, 0.600]	0.706 $\pm$ 0.092	[0.673, 0.738]
Mean	0.792 $\pm$ 0.110	[0.779, 0.805]	**0.858 $\pm$ 0.107**	**[0.846, 0.871]**	0.658 $\pm$ 0.134	[0.642, 0.673]	0.556 $\pm$ 0.262	[0.526, 0.586]	0.722 $\pm$ 0.105	[0.710, 0.734]
SYNTHETICFREQUENCY										
midFreq-highFreq	0.802 $\pm$ 0.096	[0.768, 0.836]	**0.804 $\pm$ 0.096**	**[0.770, 0.837]**	0.694 $\pm$ 0.127	[0.649, 0.739]	–	–	–	–
midFreq-highFreq _conf-midhighFreq-50-50	**0.776 $\pm$ 0.121**	**[0.733, 0.819]**	0.775 $\pm$ 0.123	[0.731, 0.818]	0.690 $\pm$ 0.114	[0.650, 0.731]	–	–	–	–
midFreq-midhighFreq	**0.820 $\pm$ 0.111**	**[0.781, 0.860]**	0.819 $\pm$ 0.111	[0.780, 0.859]	0.744 $\pm$ 0.161	[0.687, 0.801]	–	–	–	–
midFreq-midhighFreq _conf-highFreq-50-50	**0.792 $\pm$ 0.109**	**[0.754, 0.831]**	0.790 $\pm$ 0.111	[0.751, 0.830]	0.690 $\pm$ 0.108	[0.651, 0.728]	–	–	–	–
midhighFreq-highFreq	**0.791 $\pm$ 0.103**	**[0.754, 0.828]**	0.789 $\pm$ 0.103	[0.752, 0.825]	0.709 $\pm$ 0.144	[0.658, 0.761]	–	–	–	–
midhighFreq-highFreq _conf-midFreq-50-50	0.770 $\pm$ 0.128	[0.724, 0.815]	**0.772 $\pm$ 0.127**	**[0.726, 0.817]**	0.679 $\pm$ 0.116	[0.638, 0.720]	–	–	–	–
Mean	**0.792 $\pm$ 0.112**	**[0.776, 0.807]**	0.791 $\pm$ 0.112	[0.776, 0.807]	0.701 $\pm$ 0.130	[0.683, 0.719]	–	–	–	–

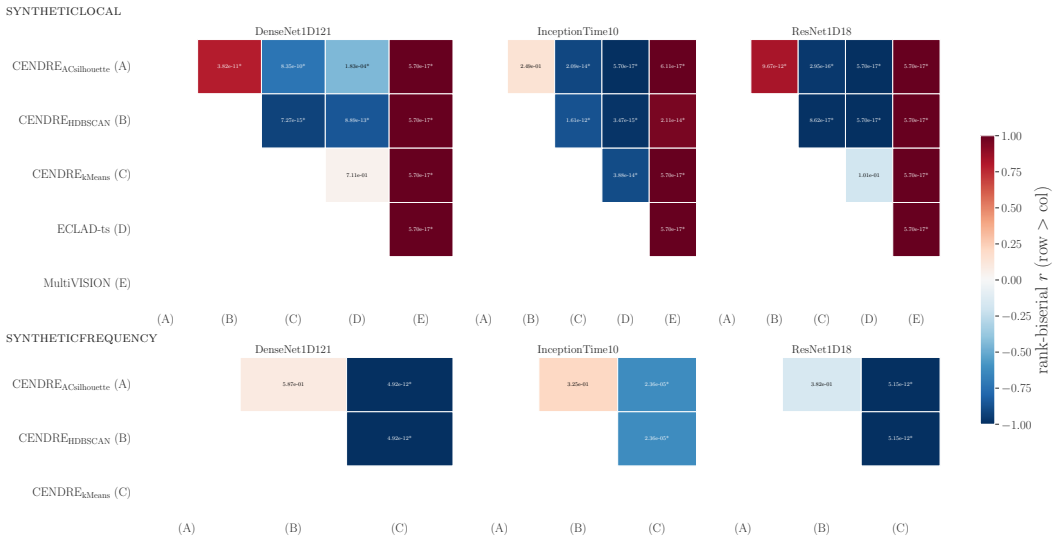
On SYNTHETICLOCAL, ECLAD-ts attains the highest sRC on all 9 datasets, with a family mean of 0.741 $[0.731, 0.751]$ separated from CENDRe$_{k\text{Means}}$'s 0.704 $[0.695, 0.714]$, the next-best method. CENDRe$_{\text{silhouette}}$ and CENDRe$_{\text{HDBSCAN}}$ trail ECLAD-ts by 0.09 and 0.11 sRC respectively, on every dataset and with comparable spread, while MultiVISION is last on sRC everywhere. The ordering reverses in sIC, where CENDRe$_{\text{HDBSCAN}}$ is best on all nine datasets (family mean 0.858), ahead of CENDRe$_{\text{silhouette}}$ (0.792) and MultiVISION (0.722), with CENDRe$_{k\text{Means}}$ at 0.658 and ECLAD-ts at 0.556. ECLAD-ts is also the least stable method under sIC, with per-dataset standard deviations up to 0.305 against 0.140 for the two automatic CENDRe variants. Its sRC lead therefore does not transfer to sIC: broader masks that cover the ground-truth primitive, selected at the K with the best sRC, do not track the model's per-timestep importance signal.

On SYNTHETICFREQUENCY, only the CENDRe variants are applicable, since the baselines have no frequency-domain counterpart, which the tables mark as "–". Both automatic variants lose about 0.06 sRC to CENDRe$_{k\text{Means}}$ and gain about 0.09 sIC, on every dataset of the family, the same trade observed on SYNTHETICLOCAL. CENDRe$_{\text{silhouette}}$ and CENDRe$_{\text{HDBSCAN}}$ behave similarly on SYNTHETICFREQUENCY (family means within 0.001 sRC and sIC of one another) though with 11 seeds their per-run scores no longer coincide everywhere, matching exactly in 42% of the (dataset, architecture, seed) combinations. Compared with SYNTHETICLOCAL, mean sRC is lower (0.510 against 0.651 for CENDRe$_{\text{silhouette}}$) while mean sIC is unchanged (0.792 in both cases). Frequency-domain concepts thus track the model's importance signal as well as time-domain ones, while matching the ground-truth band less tightly.

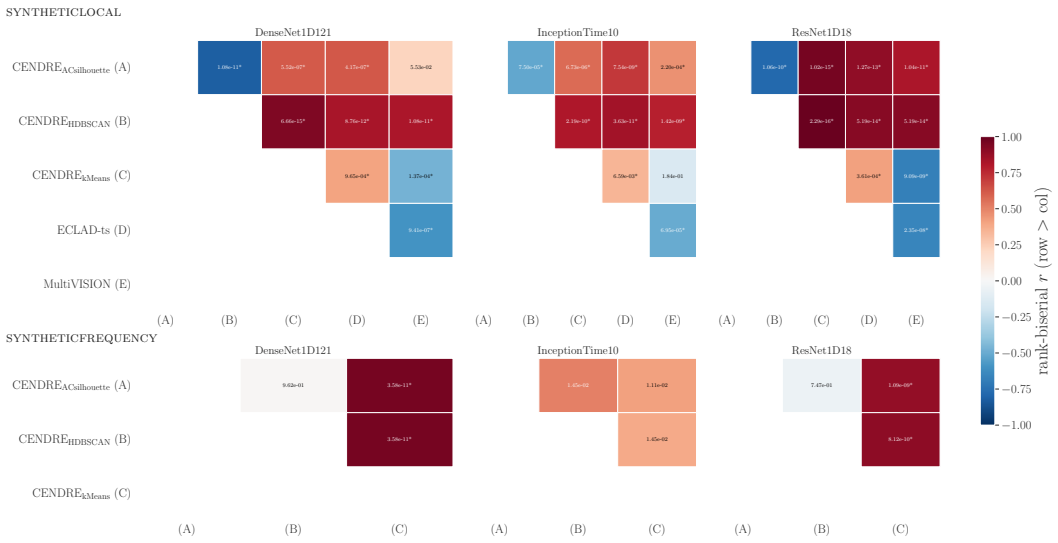
D.1.4 Quantitative evaluation: statistical significance

Section 5 compares the methods by reading the boxplots of Figure 2, where an ordering is visible but not tested. We test that ordering here, with no additional experiments. Each run is one combination of dataset, architecture, and seed, and every applicable method is evaluated on all of them, so the scores form matched pairs. We test each architecture separately over 11 seeds, giving 99 matched pairs per method pair on SYNTHETICLOCAL (9 datasets $\times$ 11 seeds) and 66 on SYNTHETICFREQUENCY

(6 datasets $\times$ 11 seeds). Runs in which $\text{CENDRe}_{\text{HDBSCAN}}$ returned no concepts are dropped from the pairs involving it. We compare each pair with a two-sided Wilcoxon signed-rank test [45], and Holm-correct the p-values over all pairs within a family, metric, and architecture [15]. We accompany each test with the matched-pairs rank-biserial correlation r [20], the share of pairs favoring the row method minus the share favoring the column method. Figure 9 reports both, with cell color giving r (red for row above column, blue for row below) and cell text the Holm-adjusted p-value. On SYNTHETICFREQUENCY, only the three CENDRe variants enter the comparison, since the baselines have no frequency-domain counterpart. Of the 78 tests in Figure 9 (10 method pairs



(a) sRC



(b) sIC

Figure 9: Pairwise Holm-corrected Wilcoxon signed-rank tests between CE methods, per architecture, on the runs of Figure 2, for (a) sRC and (b) sIC. Within each panel, the upper block is SYNTHETICLOCAL and the lower block SYNTHETICFREQUENCY, and the three column groups are the DenseNet1D-121, the InceptionTime10, and the ResNet1D-18. Only the upper triangle is shown, since the test is symmetric. Cell color gives the matched-pairs rank-biserial correlation r (red for row above column, blue for row below), and cell text the Holm-adjusted p-value, with an asterisk marking $p < 0.01$. On SYNTHETICFREQUENCY, only the three CENDRe variants enter the comparison, since the baselines have no frequency-domain counterpart.

on SYNTHETICLOCAL and 3 on SYNTHETICFREQUENCY, per architecture and metric), 65 reach $p < 0.01$, and the sign of r follows the ordering of §D.1.3 throughout. The tests therefore confirm the reading of Figure 2. The 13 pairs that do not separate concentrate in three places, with two isolated exceptions.

The first is CENDRe$_{\text{silhouette}}$ against CENDRe$_{\text{HDBSCAN}}$ on SYNTHETICFREQUENCY, which separates on neither metric nor any architecture ($p \geq 0.014$ across all six tests). This matches their family means in §D.1.3, which differ by less than 0.001 in both metrics. On SYNTHETICLOCAL the two criteria separate everywhere except in sRC on the InceptionTime10 ($p = 0.25$). Both methods select the number of centroids automatically and visualize concepts by the same means, so similar behaviour on some datasets is expected.

The second is CENDRe$_{k\text{Means}}$ against ECLAD-ts in sRC on SYNTHETICLOCAL, indistinguishable on the DenseNet1D-121 ($p = 0.71$) and the ResNet1D-18 ($p = 0.10$) but separated on the InceptionTime10. The gap of 0.035 sRC between their family means is therefore carried by a single architecture. In sIC the same pair separates on all three, with ECLAD-ts behind. The two methods share the same clustering step, so their scores differ mainly through the mode of concept visualization. The visualizations converge when the CENDRe masks are dense, which can occur on individual architecture-dataset combinations.

The third is the InceptionTime10 in sIC on SYNTHETICFREQUENCY, where no pair of CENDRe variants reaches the 1% level (p between 0.011 and 0.015). The other two architectures separate CENDRe$_{k\text{Means}}$ from both automatic variants there, so this reflects the backbone rather than the metric.

The two remaining cells both involve MultiVISION in sIC on SYNTHETICLOCAL: against CENDRe$_{\text{silhouette}}$ on the DenseNet1D-121 ($p = 0.055$) and against CENDRe$_{k\text{Means}}$ on the InceptionTime10 ($p = 0.18$). Both are consistent with MultiVISION ranking mid-field on sIC rather than last, and neither affects the ordering among the CENDRe variants or against ECLAD-ts.

D.1.5 Multichannel localization and confounder robustness

We extend the quantitative evaluation of Figure 2 to two synthetic datasets: SYNTHETICLMC (D=2, per-channel attribution) and SYNTHETICLCONF (D=1, paired with a third shape as confounder at $40/60$, $50/50$, and $60/40$ ratios). Figure 10 reports sRC and sIC across all variants and the three architectures. These two families run over 3 seeds (0–2), rather than the 11 of the main families. The ranking is similar to the main paper. CENDRe$_{\text{silhouette}}$ and CENDRe$_{\text{HDBSCAN}}$ trade slightly lower sRC for noticeably higher sIC than ECLAD-ts, CENDRe$_{k\text{Means}}$ is similar to ECLAD-ts on both metrics, and MultiVISION underperforms on sRC.

Across these settings, CENDRe$_{\text{HDBSCAN}}$ reaches slightly higher sIC than CENDRe$_{\text{silhouette}}$, consistent with density-based clustering being better suited to the non-spherical structures one expects in latent spaces. We still report silhouette-guided aggregation as the default in the main text because HDBSCAN occasionally requires per-dataset tuning of its neighborhood parameters when its single-pass run fails to find any dense region (see failure rates below); on more complex problems where this tuning effort is acceptable, HDBSCAN may be the stronger choice. CENDRe$_{\text{HDBSCAN}}$ occasionally fails to converge, with HDBSCAN labeling all 50 micro-centroids as noise, on $11/297$ SYNTHETICLOCAL and $5/54$ SYNTHETICLCONF runs (all on InceptionTime10 or ResNet1D-18, concentrated on `triangle-circle` bases). No failures occur on SYNTHETICFREQUENCY ($0/198$) or SYNTHETICLMC ($0/54$). Failed runs are excluded from the boxplot statistics, so the boxplots are conditional on convergence.

D.2 Natural datasets

We complement the CWRU results of Section 5 with two further natural-data evaluations. We extend the qualitative evaluation to five univariate datasets from the UCR archive, covering cardiac, automotive, motion, robotic, and semiconductor signals (§D.2.1). We then extract concepts from a second bearing-fault dataset, Paderborn Bearing (BearingPD), on two architectures, to test whether CENDRe recovers the same fault-relevant frequency bands under a different rig and sensor setup.

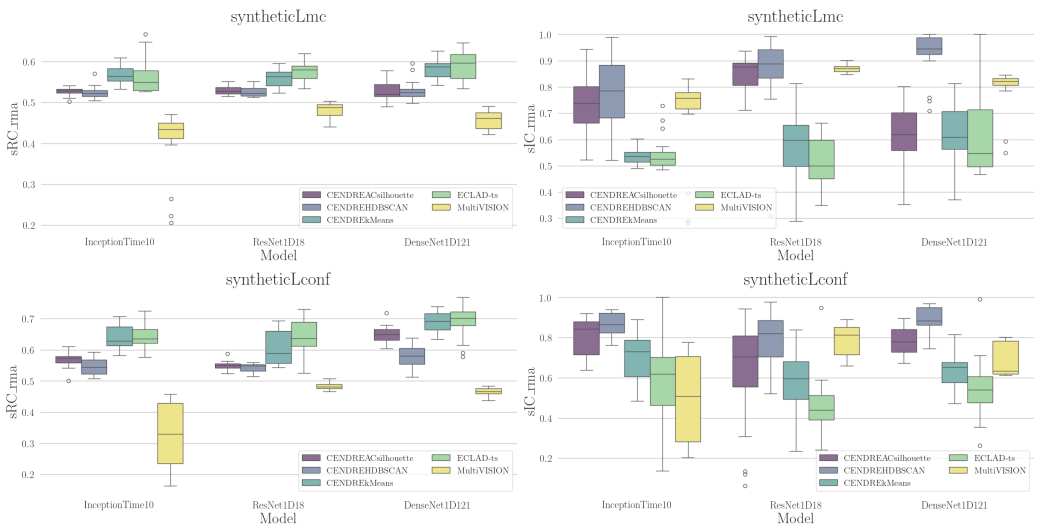


Figure 10: Quantitative evaluation on SYNTHETICLMC (top, per-channel attribution) and SYNTHETICLCONF (bottom, paired with confounders). The ranking mirrors Figure 2: $\text{CENDRe}_{\text{silhouette}}$ and $\text{CENDRe}_{\text{HDBSCAN}}$ achieve sRC close to ECLAD-ts and higher sIC, $\text{CENDRe}_{k\text{Means}}$ tracks ECLAD-ts on both metrics, and MultiVISION is weak on sRC.

D.2.1 UCR

We extend the qualitative evaluation to five univariate datasets of the UCR archive [9], covering cardiac, automotive, motion, robotic, and semiconductor signals. For each dataset we train the three architectures of Section 4.2 and extract concepts from one of them, using $\text{CENDRe}_{\text{silhouette}}$ with the hyperparameters of Section 4.2. Each report shows the time-domain masks, the frequency-domain masks, and the global concept-frequency correspondence. The archive distributes most of its datasets without the sampling rate of the original recording. Where the rate is undocumented we plot the frequency axis in cycles per sample, under $f_s = 1$, rather than assume a rate. Where it is documented we state it in the corresponding paragraph and label the time and frequency axes in seconds and Hz. The choice affects the labeling of the axes alone, since the masks are computed on the sampled series either way.

ECG200 [9] traces the electrical activity recorded during a single heartbeat, on one electrocardiogram channel. The two classes are a normal heartbeat and a myocardial infarction. Series have $T = 96$ samples. $\text{CENDRe}_{\text{silhouette}}$ extracts $K = 4$ concepts, which partition the beat into consecutive segments. C_0 and C_1 carry the largest and opposite-signed importance, with C_0 activating on the plateau that follows the main negative deflection, mostly in the $y = 1$ series, and C_1 on the deflection itself. C_2 and C_3 cover the late and the early part of the beat at lower importance. In frequency, all four concepts keep their mass below 0.25 cycles per sample and peak near 0.02, so the model reads the morphology of the beat rather than high-frequency content.

FordA [9] contains engine noise measurements from an automotive subsystem, with the task of deciding whether a given symptom is present. Series have $T = 500$ samples, recorded under typical operating conditions and with minimal contamination. $\text{CENDRe}_{\text{silhouette}}$ extracts $K = 2$ concepts with opposite class importance. C_0 concentrates on short high-amplitude bursts and C_1 on the remaining oscillation, the two alternating along the window instead of splitting it into contiguous segments. Both confine their frequency mass to the band below 0.15 cycles per sample, which carries the dominant spectral peak of the signal, and both peak near 0.05. Within that band the two profiles differ in shape: C_0 concentrates its mass around the 0.05 peak and decays quickly above it, while C_1 spreads a broader shoulder that persists toward 0.15.

GunPoint [9] tracks the horizontal position of the centroid of an actor's right hand, extracted from video of one of two motions. In the gun class the actor draws a replica gun from a hip holster, points it at a target for about one second, and returns it. In the point class the actor performs the same motion

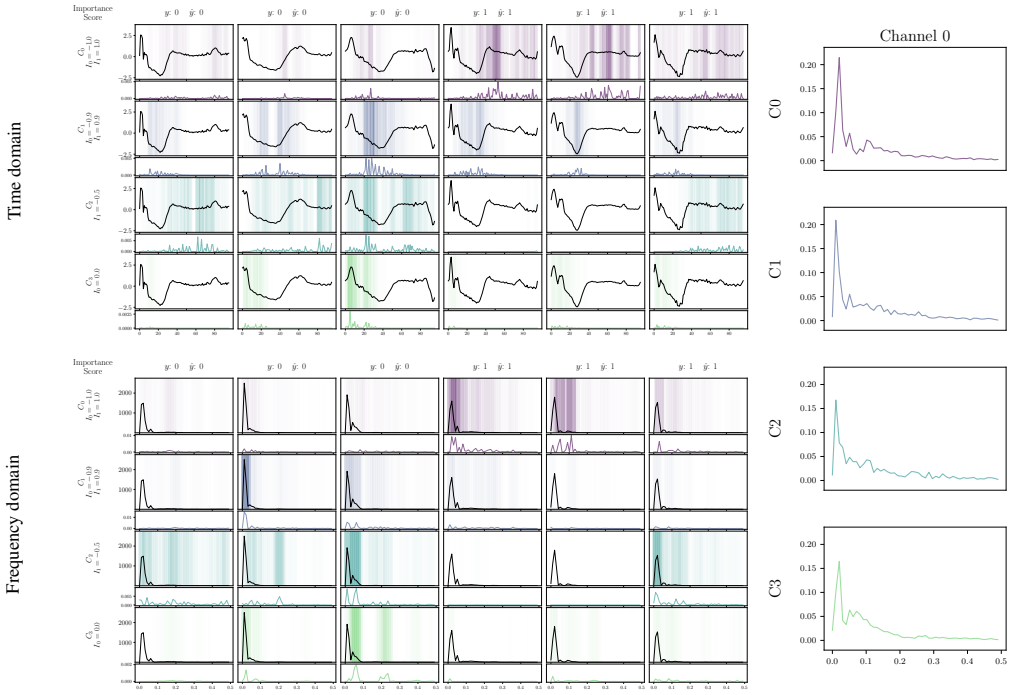


Figure 11: Concept extraction from a DenseNet1D-121 trained on ECG200 (two classes: normal heartbeat, myocardial infarction), reaching 91.67% test accuracy. Top left: time-domain concept masks. Bottom left: frequency-domain concept masks. Right: global concept-frequency correspondence. The concepts split the beat into consecutive segments, and none of them reads content above 0.25 cycles per sample.

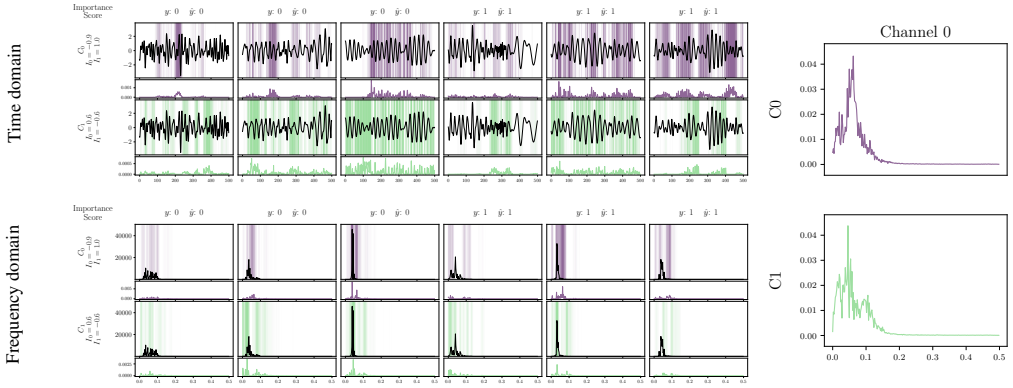


Figure 12: Concept extraction from an InceptionTime10 trained on FordA (two classes: symptom present, symptom absent), reaching 92.48% test accuracy. Top left: time-domain concept masks. Bottom left: frequency-domain concept masks. Right: global concept-frequency correspondence. Both concepts peak near 0.05 cycles per sample; C_0 concentrates its mass around that peak, while C_1 spreads a broader shoulder across the band below 0.15.

with an extended index finger and no prop. Series have $T = 150$ samples from two actors. The motion was recorded at 30 Hz, so the axes are labeled in seconds and in Hz over $[0, 15]$. CENDRe$_{\text{silhouette}}$ extracts $K = 4$ concepts, all of which localize on the transitions of the hand trajectory and none on the plateau where the hand is held on target. C_0 and C_1 cover the rising edge around 1–2 s and C_2 the falling edge around 3–4 s in the $y = 0$ series, while C_3 covers both edges in the $y = 1$ series. Every concept peaks between 0.3 and 0.6 Hz and carries no mass above 2 Hz, on the order of the one-second gesture. The extraction recovers the draw and the return as separate concepts, which is the distinction the two motions are defined by, as has been observed in literature [14].

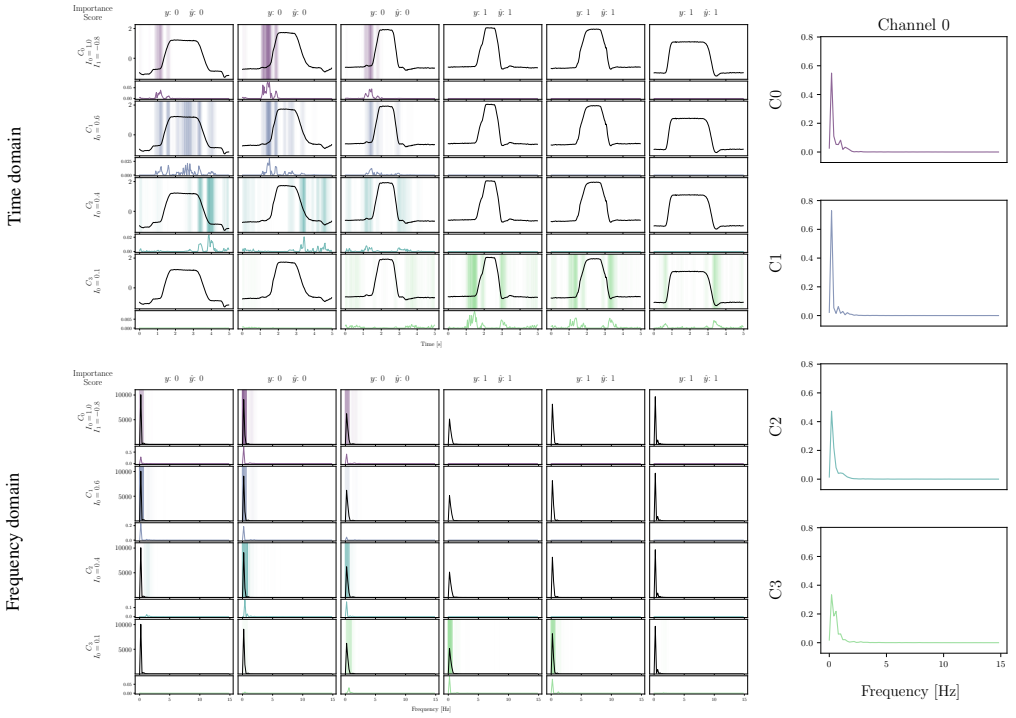


Figure 13: Concept extraction from an InceptionTime10 trained on GunPoint (two classes: gun draw, point), reaching 100% test accuracy. Top left: time-domain concept masks. Bottom left: frequency-domain concept masks. Right: global concept-frequency correspondence. All four concepts localize on the draw and the return of the hand, and none on the plateau in between.

SonyAIBORobotSurface1 [9] contains accelerometer readings from a Sony AIBO robot, taken while the robot walks on one of two surfaces. The two classes are cement and carpet. The robot carries roll, pitch, and yaw accelerometers, of which the archive distributes the x axis alone. Series have $T = 70$ samples. The accelerometer samples at 125 Hz, so the axes are labeled in seconds and in Hz over $[0, 62.5]$. CENDRe$_{\text{silhouette}}$ extracts $K = 2$ concepts that separate by class rather than by band. C_0 activates on the $y = 1$ series and C_1 on the $y = 0$ series, each spread over the whole 0.55 s window instead of a contiguous segment. Both concentrate their mass between 5 and 25 Hz, peak near 8 Hz, and decay above 30 Hz, in the range of the step cadence of the walk. Within that band the two profiles differ in shape: C_0 splits its mass into two sharp peaks near 8 and 14 Hz, while C_1 pairs the 8 Hz peak with a lower but longer ridge that extends to about 30 Hz.

Wafer [9] contains inline process control measurements from semiconductor fabrication, each series recorded by one sensor while one tool processes one silicon wafer. The two classes are normal and abnormal, with abnormal covering 10.7% of the training and 12.1% of the test series. Series have $T = 152$ samples. CENDRe$_{\text{silhouette}}$ extracts $K = 5$ concepts. Four of them cover consecutive stages of the process trace, with C_1 and C_4 on the initial transient, C_3 on the plateau, and C_2 on the final ramp. The fifth, C_0, carries the largest importance magnitude with opposite sign across the two classes and spans the whole window of the $y = 1$ series. The frequency masks are broader than on the other four datasets: C_2, C_3, and C_4 peak below 0.03 cycles per sample, while C_0 and C_1 cover the full band, C_1 as a near-uniform comb. That comb is the signature of the step-like transients of the trace rather than of a narrowband feature, which is what a purely spectral summary of this model would misread.

D.2.2 BearingPD

We complement the CWRU example in the main text with concept extraction on a second natural bearing-fault dataset. *Paderborn Bearing (BearingPD)* [25] contains accelerometer recordings from a motor test rig at Paderborn University, covering healthy bearings and bearings with artificially

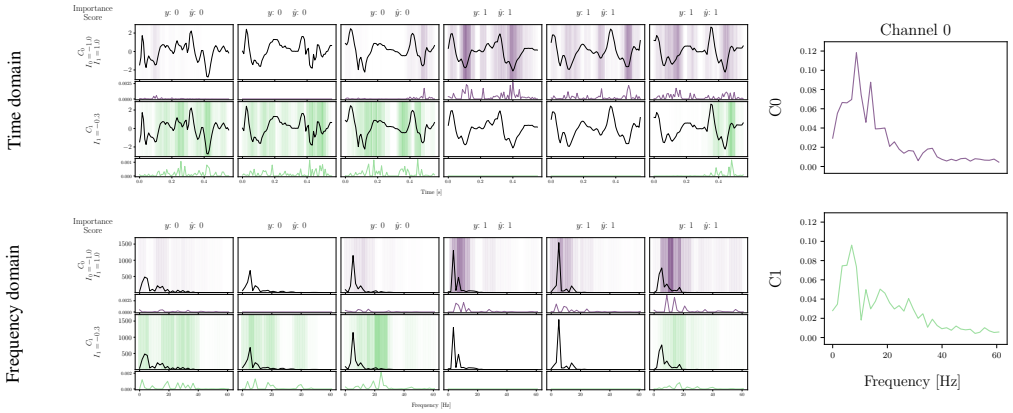


Figure 14: Concept extraction from a DenseNet1D-121 trained on SonyAIBORobotSurface1 (two classes: cement, carpet), reaching 97.50% test accuracy. Top left: time-domain concept masks. Bottom left: frequency-domain concept masks. Right: global concept-frequency correspondence. Both concepts peak near $8\,\mathrm{Hz}$ within the gait band; C_0 adds a second sharp peak near $14\,\mathrm{Hz}$, while C_1 spreads a lower ridge up to about $30\,\mathrm{Hz}$.

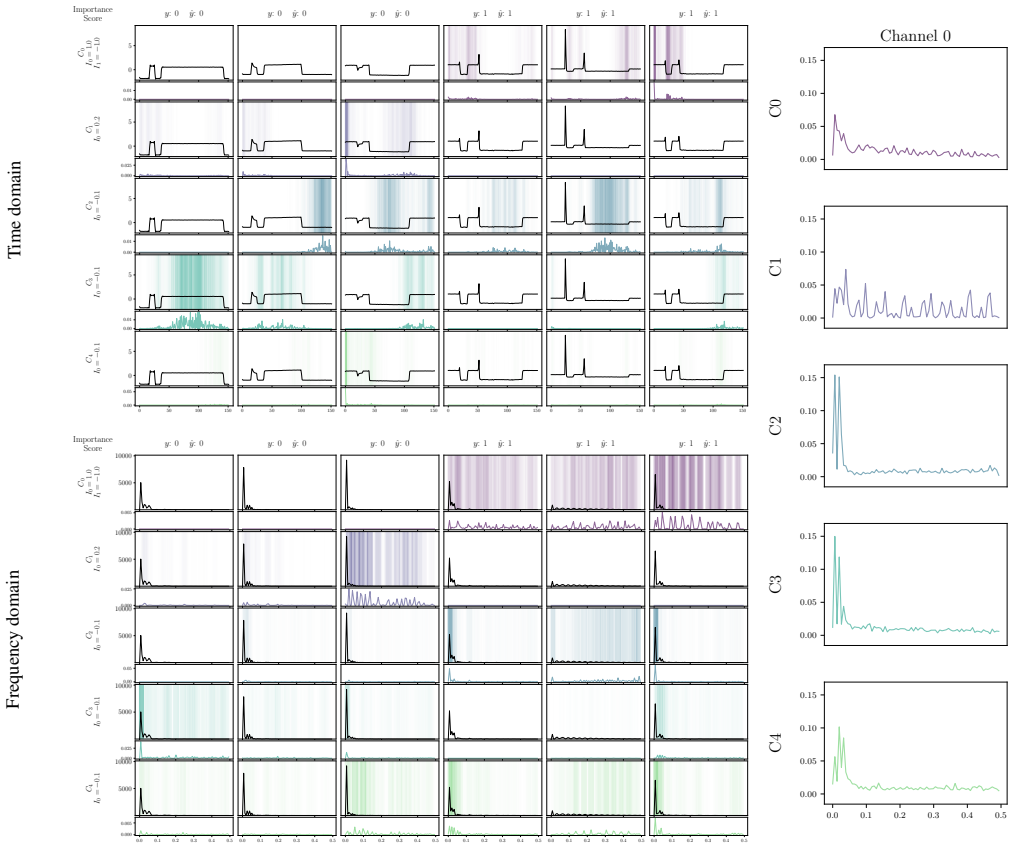


Figure 15: Concept extraction from a ResNet1D-18 trained on Wafer (two classes: normal, abnormal), reaching 100% test accuracy. Top left: time-domain concept masks. Bottom left: frequency-domain concept masks. Right: global concept-frequency correspondence. Four concepts follow the stages of the process trace, while C_0 spans the whole window and carries the strongest class-discriminative importance.

induced or naturally degraded inner and outer race damage; we use a three-class setup (healthy, inner-race fault, outer-race fault). Recordings are sampled at $f_s = 64\,\mathrm{kHz}$ on a single vibration

channel and segmented into non-overlapping windows of $T = 5000$ samples. Training follows Section 4.2, with two additions: Inputs are standardized by the global mean and standard deviation of the training set, and Gaussian noise of standard deviation 0.01 is added to the standardized signal during training as augmentation.

On BearingPD on the InceptionTime10 model (Figure 16a), the extracted concepts partly separate by class, though some concepts activate in more than one class with differing activation strengths. The frequency-domain masks and the global concept-correspondence concentrate concept mass on four distinguishable bands, with markedly different distributions across concepts: C_0 spans all four, with its largest mass at $0.6\,\text{kHz}$ and comparable mass in the 2 and $3.4\,\text{kHz}$ bands; C_1 and C_2 concentrate on the low band, peaking at $1.3\,\text{kHz}$, and retain only weak activity at 2 and $3.4\,\text{kHz}$; all three carry a narrow peak near $10\,\text{kHz}$, and none shows a distinct peak above it. All four bands fall inside the structural-resonance region typically used for envelope analysis on this kind of bearing rig [35], and the per-concept differences within that region reflect that the model has partitioned the resonance content into class-specific sub-bands rather than treating it as a single feature. These specific bands give domain experts the information needed to assess whether the model is reasoning over fault-relevant spectral content, and the agreement with the literature regions supports that CENDRe transfers from controlled synthetic settings to real signals.

On BearingPD on the DenseNet1D-121 model (Figure 16b), CENDRe recovers the same four bands, distributed differently across the concepts. The concepts again separate only partly by class, with some activating in more than one class at differing strengths. C_0 places its largest mass at $3.4\,\text{kHz}$, with the low pair close behind and a clear peak near $10\,\text{kHz}$; C_1 is dominated by a sharp peak at $1.3\,\text{kHz}$, roughly twice the height of any peak in the other two concepts; C_2 carries the 0.6 and $1.3\,\text{kHz}$ peaks at comparable height, with weaker mass at 3.4 and $10\,\text{kHz}$. Both architectures resolve the low band into the same 0.6 and $1.3\,\text{kHz}$ pair, and in both, one concept is dominated by a single member of the pair while the others carry both at comparable height, though the concept indices playing each role differ. Just as for InceptionTime10, these bands give domain experts the information needed to assess whether the model is reasoning over fault-relevant spectral content. Recovering them under a second architecture supports that CENDRe transfers from controlled synthetic settings to real signals independently of the backbone.

E Hyperparameter sensitivity

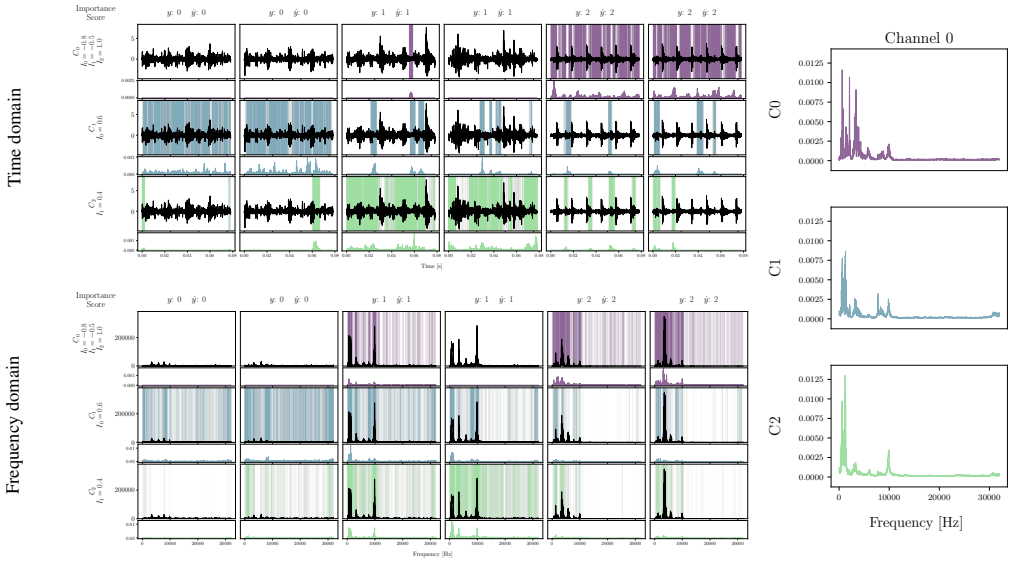
The $\text{CENDRe}_{\text{silhouette/HDBSCAN}}$ variants replace the number of concepts K with the number of micro-clusters J. We investigate the sensitivity of CENDRe to J by sweeping J with every other hyperparameter fixed at its main-experiment value. The grid is bounded by the method itself: the upper bound is the number of LADs in one mini-batch ($J_{\max} = 3200$ in our configuration), since k-means cannot form more clusters than it sees in one partial fit, and the lower bound $J_{\min} = 11$ follows from the agglomerative candidate search over up to 10 concepts. Within $[J_{\min}, J_{\max}]$ we sample 10 log-spaced values, snapped to include the main-experiment value $J = 50$, giving $J \in \{11, 21, 50, 73, 137, 257, 483, 907, 1704, 3200\}$. The sweep covers two datasets from each of the main synthetic families and the 11 seeds of the main experiments, reused at every level of J so that seed acts as a block rather than an independent replicate. Runs in which the extractor returned no concepts are excluded, which affects $\text{CENDRe}_{\text{HDBSCAN}}$ at small J only.

Figure 17 shows the seed mean and standard deviation of sRC, sIC and K per dataset. $\text{CENDRe}_{\text{silhouette}}$ is flat over three orders of magnitude of J on all four datasets, except at the two smallest values, where too few micro-clusters remain for the candidate search. $\text{CENDRe}_{\text{HDBSCAN}}$ matches it on SYNTHETICFREQUENCY, but on SYNTHETICLOCAL its sIC collapses above a few hundred micro-clusters, with a widening seed spread.

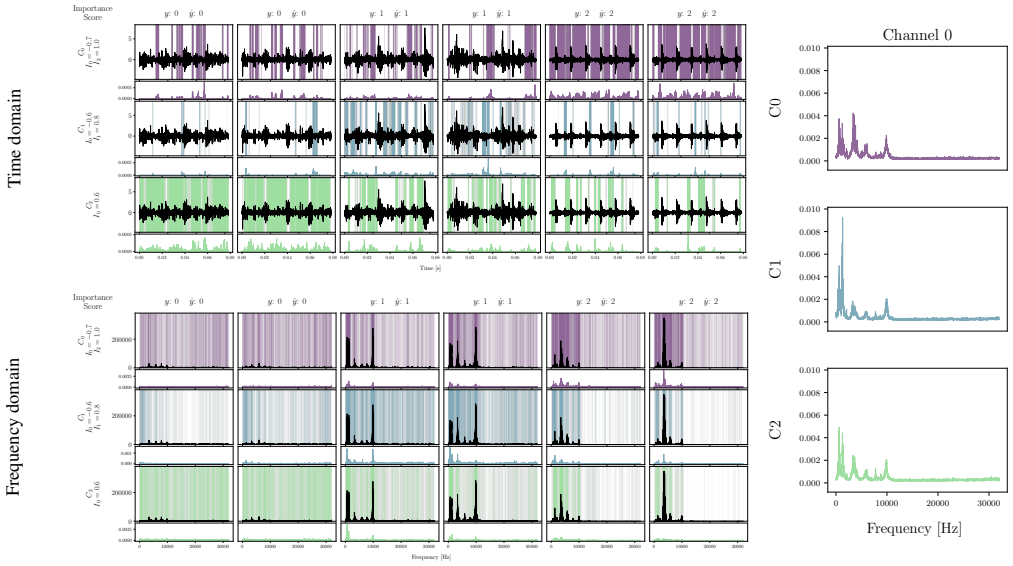
We quantify the sensitivity with two standard measures, both taken under a log-uniform distribution of J on $[J_{\min}, J_{\max}]$, which the log-spaced grid samples uniformly [4]. Writing the score as $y(J, \varepsilon)$, where ε collects everything the seed controls, the first-order Sobol index [42, 38]

$$S_1^J = \frac{\text{Var}_J(\mathbb{E}_\varepsilon[y \mid J])}{\text{Var}(y)} \tag{9}$$

is the share of the total variance explained by the choice of J, and S_1^{seed} exchanges the roles of J and ε. We estimate Equation 9 from the paired design as a two-way variance decomposition over



(a) InceptionTime10, where C_0 carries the 2 and 3.4 kHz bands alongside the low pair, while C_1 and C_2 concentrate on the low pair.



(b) DenseNet1D-121, where C_0 peaks at 3.4 kHz, C_1 is dominated by a sharp 1.3 kHz peak, and C_2 carries the 0.6–1.3 kHz pair at comparable height.

Figure 16: Concept extraction on BearingPD (three classes: healthy, inner-race fault, outer-race fault) for two architectures, each trained with seed 0 to 100% validation accuracy. Top left of each panel: time-domain concept masks, which localize between high-amplitude transients and pick up patterns that align with the periodic impact structure of the signal. Bottom left: frequency-domain concept masks, whose mass concentrates on four bands at roughly 0.6–1.3, 2, 3.4, and 10 kHz. Right: global concept-frequency correspondence per channel, where both architectures show a peak near 10 kHz. The two architectures recover the same four bands, all inside the structural-resonance region commonly used for envelope-based bearing diagnostics [35], and differ only in how the bands distribute across concepts.

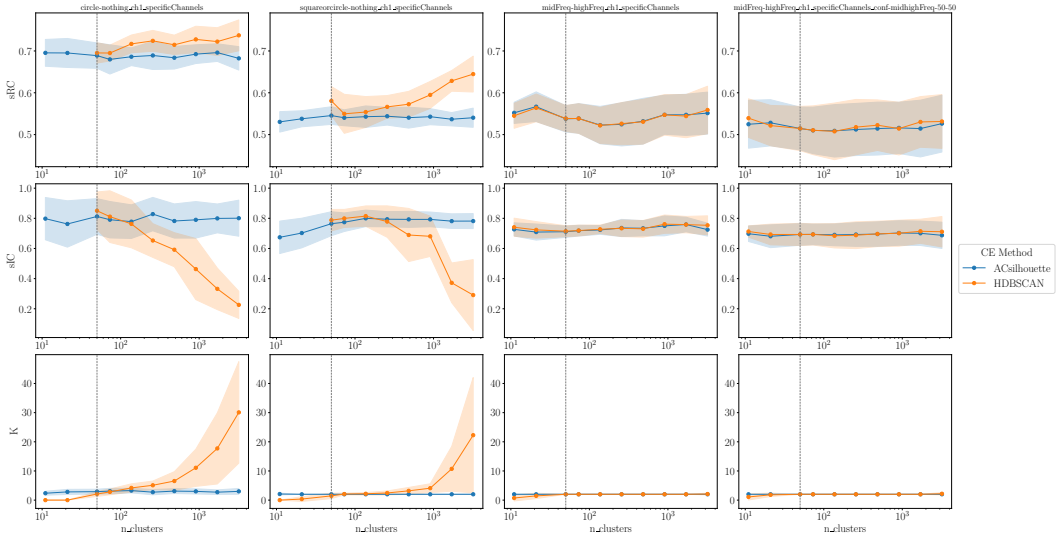


Figure 17: Response of sRC (top row), sIC (middle row) and K (bottom row) to the number of micro-clusters J, per dataset, for $\text{CENDRe}_{\text{silhouette}}$ and $\text{CENDRe}_{\text{HDBSCAN}}$. Markers are the mean over 11 seeds and shading is one standard deviation, with all panels on a common y-range. The dashed vertical line is the value $J = 50$ used in the main experiments.

levels of J and seeds, in the bias-corrected ω^2 form [30], weighting the levels by their log spacing. A variance share carries no direction, so we add the elementary effects of Morris [28]. Taking the factor as $\log_2 J$, which the geometric grid samples at near-uniform steps, the effect of the step from J_j to J_{j+1} under seed s is

$$EE_{s,j} = \frac{y_{s,j+1} - y_{s,j}}{\log_2 J_{j+1} - \log_2 J_j}. \tag{10}$$

Their mean μ_{EE} reads as the average change in the score per doubling of J, in the units of that score. We report the signed mean rather than the absolute-value variant μ^* [7], since with a single swept factor the sign carries the direction of the drift.

Tables 7a, 7b, and 7c report the three measures. For $\text{CENDRe}_{\text{silhouette}}$, J explains on average 0.022 of the variance in sRC, 0.072 in sIC, and 0.010 in the selected K itself, against 0.702, 0.577, and 0.159 for the seed. The mean elementary effect on K is a negligible $+0.02$ concepts per doubling of J, consistent with the near-zero effect already observed on sRC and sIC. Choosing J is therefore a smaller decision than the run-to-run variation the method already tolerates, for the score and for K alike. $\text{CENDRe}_{\text{HDBSCAN}}$ is more sensitive on average in sRC and sIC ($S_1^J = 0.144$ and 0.320). Its selected K tracks J far more directly, with $S_1^J = 0.416$ on average and a mean elementary effect reaching $+3.67$ concepts per doubling of J on CIRCLE-NOTHING. This uncontrolled growth in K plausibly explains the sIC collapse noted above. Sensitivity to J, in the score and in the selected K, is therefore a property of the density-based cluster mode rather than of CENDRe itself. The one exception is sIC on `squareorcircle-nothing`, where the silhouette variant also reaches $S_1^J = 0.23$.

F Computational resources

The experiments span two systems, an institutional HPC cluster and a local workstation. Each HPC GPU node provides two Intel Xeon CPUs (24 physical cores each, 192 GB RAM), four NVIDIA A100 (40 GB) GPUs, and an InfiniBand interconnect. The workstation has dual Intel Xeon Gold 6330 CPUs (2×28 cores, 112 logical threads at 2.00 GHz, 1 TiB RAM) and two NVIDIA A100-PCIE-40GB GPUs.Model training ran entirely on the cluster. The CE and synthetic-metric experiments ran on both systems, so their runtimes are reported as wall-clock measurements taken on the workstation under controlled conditions and aggregated over the full sweep.

Table 7: Sensitivity to the number of micro-clusters J, for sRC, sIC, and K. The columns S_1^J and S_1^{seed} are the first-order Sobol indices of Equation 9 for J and for the random seed, and μ_{EE} is the mean elementary effect of Equation 10, in units of the reported score per doubling of J. The final row of each panel averages each index over datasets.

(a) Sensitivity of sRC to J.

	CENDRe$_{\text{silhouette}}$			CENDRe$_{\text{HDBSCAN}}$		
Dataset	S_1^J	S_1^{seed}	μ_{EE}	S_1^J	S_1^{seed}	μ_{EE}
circle-nothing	0.009	0.666	-0.0021	0.081	0.383	+0.0078
squareorcircle-nothing	0.008	0.696	+0.0005	0.448	0.045	+0.0174
midFreq-highFreq	0.069	0.591	+0.0009	0.036	0.746	+0.0032
midFreq-highFreq _conf-midhighFreq-50-50	0.000	0.854	+0.0003	0.012	0.910	+0.0022
Mean	0.022	0.702	-0.0001	0.144	0.521	+0.0076

(b) Sensitivity of sIC to J.

	CENDRe$_{\text{silhouette}}$			CENDRe$_{\text{HDBSCAN}}$		
Dataset	S_1^J	S_1^{seed}	μ_{EE}	S_1^J	S_1^{seed}	μ_{EE}
circle-nothing	0.000	0.633	-0.0029	0.599	0.047	-0.1073
squareorcircle-nothing	0.233	0.479	+0.0118	0.640	0.094	-0.0932
midFreq-highFreq	0.056	0.290	+0.0003	0.041	0.207	+0.0066
midFreq-highFreq _conf-midhighFreq-50-50	0.000	0.907	-0.0016	0.000	0.729	+0.0029
Mean	0.072	0.577	+0.0019	0.320	0.269	-0.0478

(c) Sensitivity of K to J.

	CENDRe$_{\text{silhouette}}$			CENDRe$_{\text{HDBSCAN}}$		
Dataset	S_1^J	S_1^{seed}	μ_{EE}	S_1^J	S_1^{seed}	μ_{EE}
circle-nothing	0.041	0.635	+0.0880	0.589	0.041	+3.6664
squareorcircle-nothing	0.000	0.000	-0.0108	0.470	0.001	+2.7356
midFreq-highFreq	0.000	0.000	+0.0000	0.356	0.026	+0.1462
midFreq-highFreq _conf-midhighFreq-50-50	0.000	0.000	+0.0000	0.248	0.010	+0.1250
Mean	0.010	0.159	+0.0193	0.416	0.020	+1.6683

F.1 Model training

Model training was performed on 3 architectures over 27 dataset configurations, with 11 seeds on the 15 SYNTHETICLOCAL and SYNTHETICFREQUENCY datasets and 3 seeds on the 12 SYNTHETICLMC and SYNTHETICLCONF datasets, totaling 603 trained models. Training consumed a total of 40.64 GPU hours.

F.2 Concept extraction and synthetic metrics

Runtimes below are wall-clock measurements collected on the workstation from controlled batches on a single partition with 25 allocated CPUs and no concurrent GPU users. Both CE and synthetic-metric runtimes are reported per (architecture, method) as the mean over 3 seeds on the SYNTHETICLOCAL square-triangle variant, summed over the K-sweep used in the comparison. For the AC-clustered CENDRe variants (silhouette, HDBSCAN), K is determined in a single pass, so a single run replaces the sweep, and the tables reflect this (they are 6–9$\times$ faster than the fixed-K baselines). The experiments on ECLAD-ts and MultiVISION were done on the SYNTHETICLOCAL family (9 datasets, 11 seeds) and the SYNTHETICLMC and SYNTHETICLCONF families (12 datasets, 3 seeds), which correspond to $9 \times 11 + 12 \times 3 = 135$ runs per architecture. The CENDRe variants additionally run on the SYNTHETICFREQUENCY family (6 datasets, 11 seeds), making for a total of $135 + 66 = 201$ runs per architecture.

Concept extraction. Aggregated over the full sweep, CE consumed 68.54 GPU hours. A further 29.97 GPU hours were spent on exploratory and repeated runs during development, an amount that does not grow with the seed count.

Table 8: Mean CE runtime per (architecture, method), in seconds, over 3 seeds on a representative SYNTHETICLOCAL variant. Fixed-K methods (ECLAD-ts, MultiVISION, CENDRe$_{k\text{Means}}$) sum over $K \in \{2, 4, 6, 8, 10\}$; AC variants run once.

Architecture	ECLAD-ts	MultiVISION	CENDRe$_{k\text{Means}}$	CENDRe$_{\text{silhouette}}$	CENDRe$_{\text{HDBSCAN}}$
InceptionTime10	102	120	118	18	15
ResNet1D-18	88	103	91	11	11
DenseNet1D-121	251	321	227	40	35

Synthetic metrics. The representation-correctness (RC) and importance-correctness (IC) metrics are computed jointly in a single pass, so the table reports one runtime per (architecture, method) covering both metrics together. Aggregated over the full sweep, synthetic-metric computation con-

Table 9: Mean synthetic-metric runtime (RC and IC computed jointly) per (architecture, method), in seconds, over 3 seeds on a representative SYNTHETICLOCAL variant. Fixed-K methods sum over $K \in \{2, 4, 6, 8, 10\}$; AC variants run once.

Architecture	ECLAD-ts	MultiVISION	CENDRe$_{k\text{Means}}$	CENDRe$_{\text{silhouette}}$	CENDRe$_{\text{HDBSCAN}}$
InceptionTime10	108	74	333	42	29
ResNet1D-18	107	55	308	23	23
DenseNet1D-121	111	72	464	77	50

sumed 95.08 GPU hours. As for CE, an additional 39.58 GPU hours were spent during development, independent of the seed count.

In total, across model training, CE and synthetic-metric experiments, the computational resources used to produce this paper amount to approximately 274 GPU hours.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and the introduction state the three limitations of prior time-series CE methods that CENDRe addresses, and the contributions are matched by the empirical results in Section 5 and the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Scope and assumptions are stated where they appear: the method targets 1D-CNN time-series classifiers (Section 3), evaluation relies on synthetic primitives for ground-truth metrics (Section 4), and HDBSCAN convergence failures and qualitative-only assessment on natural data are discussed in App. D.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: A new algorithm for CE is proposed and tested empirically on several datasets. The closest thing to a theorems or proof would be the derivation of the frequency mask formula, which is in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides detailed descriptions of the experimental setup and results, allowing for reproduction of the main findings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and dataset generator will be made available via ZIP file, and will be released on GitHub upon publication. Instructions for reproducing the main experimental results will be included in the README file of the code repository, and a link to the repository will be provided in the final version of the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.2 specifies most training parameters, and the rest are specified in App. C, and dataset construction in App. B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Each model and CE configuration is run over 11 seeds (3 for the appendix-only synthetic families), and the variability across seeds, datasets, and architectures is shown as boxplots of sRC and sIC in Fig. 2 of Section 5.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Our experiments run on a workstation-class GPU and on our HPC, with per-model training and the full sRC/sIC sweep over $K \in \{2, 4, 6, 8, 10\}$ completing within a few hours per (model, dataset) pair; hardware, per-run timings and estimated use of resources are reported in App. F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses synthetic data and two public bearing-fault benchmarks, involves no human subjects or personally identifying data, and we have reviewed the NeurIPS Code of Ethics and find no points of conflict.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The contribution is discussed as being a general post-hoc concept-extraction algorithm for time-series CNNs, with a positive impact in enabling more faithful concept-level explanations for safety-relevant classifiers. It has no negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases a concept-extraction algorithm and evaluation code on two public benchmarks; it does not release pretrained generative models, scraped data, or any artefact with a high misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The CWRU Bearing dataset [41] and the BearingPD dataset [25] are used under their original public-research licenses, and the InceptionTime [18], ResNet1D [13], and DenseNet1D [17] reference implementations are reused under their respective open-source licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper releases the CENDRe reference implementation and the synthetic dataset (zip on submission, GitHub upon publication).

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper involves no human subjects or crowdsourced annotation: all evaluation is on synthetic data with programmatic ground-truth masks and on two public bearing-fault benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper involves no human subjects, so no IRB or equivalent approval is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the proposed method or experimental pipeline.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.